

Emacs简介

- Emacs即Editor MACroS,是一种文本编辑器
- 最初由Richard Stallman于1975年在MIT完成







Emacs简介(续)

- 自诞生以来, Emacs演化出了众多分支
- 1984年由Richard Stallman发起并由他维护至2008年的
 GNU Emacs
- 1991年发起的XEmacs
- 它们都使用了<u>Emacs Lisp</u>这种有着极强扩展性的编程语言
- 在Unix文化里,Emacs是黑客们关于编辑器优劣之争的两 、大主角之一,它的对手是vi。

Emacs的强大功能

- 收发电子邮件
- 通过FTP/TRAMP编辑远程档案
- 通过Telnet登录主机
- 上新闻组
- 对多种编程语言的编辑
- 调试程序,结合GDB,EDebug等,支持C/C++,Perl,Python,Lisp等等
- 玩游戏
- 管理日程, Task, ToDo, 约会等
- 个人信息管理
- 文件比较
- 阅读info和man文档
- 浏览网站
- · 为各种程序(<u>TeX</u>等)提供统一的操作界面

Emacs的简称

Emacs是什么的简称?(又是一个黑客幽默:) E. M. A. C. S. Emacs Makes A Computer Slow Escape Meta Alt Control Shift Emacs Makers Are Crazy Sickos Emacs Makes All Computing Simple Emacs Makefiles Annihilate C-Shells Emacs Manuals Always Cause Senility Emacs May Allow Customized Screwups Emacs Manuals Are Cryptic and Surreal Eventually Munches All Computer Storage Eight Megabytes And Constantly Swapping Elsewhere Maybe All Commands are Simple Excellent Manuals Are Clearly Suppressed Emacs May Alienate Clients and Supporters Except by Middle Aged Computer Scientists Extended Macros Are Considered Superfluous Every Mode Accelerates Creation of Software Each Manual's Audience is Completely Stupefied Exceptionally Mediocre Algorithm for Computer Scientists

Easily Maintained with the Assistance of Chemical Solutions

Eradication of Memory Accomplished with Complete Simplicity

emacs视图

```
TAGS C Senator ECB Cscope | File Edit Options Buffers Tools Search Help
                                                                                                                                                                                                                                                      /work/quagga/ospfd/
                                                                                                                                                                                                                                                    Makefile.in
          isisd =
          -lib
                                                            static int
                                                                                                                                                                                                                                                    sopf_abr.c
                                                                                                                                                                                                                                                    aspf_abr.h
          die m4
                                                            ospf db_desc_timer (struct thread *thread)
          ospf6d
                                                                                                                                                                                                                                                    - ospf_api.c
          ospfclient
                                                                                                                                                                                                                                                    ospf_api.h
                                                                struct ospf interface *oi;
                                                                                                                                                                                                                                                    ospf apiserver.c
                                                                struct ospf neighbor *nbr;
          pkgsrc
                                                                                                                                                                                                                                                    ospf apiserver.h
                                                                                                                                                                                                                                                    spf asbr.c
          a quagga-0.99.18
                                                                nbr - THREAD ARG (thread);
                                                                                                                                                                                                                                                    aspf asbr.h
          = redhat
                                                                nbr -> t db desc - NULL;
                                                                                                                                                                                                                                                    spf ase.c
          - ripd
          ripngd
                                                                                                                                                                                                                                                    aspf_ase.h
          m solaris
                                                                                                                                                                                                                                                     •
                                                                                                                                                                                                                                                    ospf dump h
          - tests
                                                                if UIS DEBUG OSPF (nsm, NSM TIMERS))
 W-0 /work/quagga/ospfd
                                                                                                                                                                                                                                                    as of flood.c
                                                                  zlog (NULL, LOG_DEBUG, "NSM[%s:%s]: Timer (DD Retransmit timer expire)",
 & ospf_neighbor.c
                                                                                                                                                                                                                                                    - ospf_flood.h
 & ospf neighbor.h
                                                                                                                                                                                                                                                    ospf_ia.c
 h ospf neighbor.lo
                                                                                                                                                                                                                                                    ospf ia.h
 & ospf network.c
                                                                                                                                                                                                                                                    es ospf interface.c
 & ospf network.h
                                                                                                                                                                                                                                                    so ospf interface h
                                                                ospf db desc resend (nbr);
 h ospf network.lo
                                                                                                                                                                                                                                                    ospf ism.c
                                                                                                                                                                                                                                                    sopf ism.h
 & ospf_nsm.h
                                                                                                                                                                                                                                                    spf_lsa.c
                                                                OSPF NSM TIMER ON (nbr->t db desc, ospf db desc timer, nbr->v db desc);
 h ospf_nsm.lo
                                                                                                                                                                                                                                                    - ospf lsa.h
 & ospf_opaque.c
                                                                                                                                                                                                                                                    ospf_lsdb.c
                                                                return O:
 E ospf opaque.h
                                                                                                                                                                                                                                                    ospf lsdb.h
                                                                                                                                                 SVN-234 (C/lh Flymake:1/0 doxy hl-p pair company AC Al " ospf main c
h ospf opaque.lo
                                                             --:--- ospf_nsm.c
                                                                                                           7% of 29k (78,8)
                                                            - Character
                                                                                                                                                                                                                                                    aspf neighbor.c
& ospf packet.c
                                                                                                                                                                                                                                                    e ospf neighbor.h
                                                                                         cscope.files
                                                                                                                                                                                 update-autotools
 ♠nsm_negotiation_done (nb)
                                                                                                                            Makefile
                                                                                                                                                    readMe.txt
                                                                                                                                                                                                                                                    espf network.c
                                                                                                                                                                                                                                                    spf network h
#nsm_notice_state_change -
#nsm_packet_received (nbr -
#nsm_should_adj (nbr : strue
                                                                                                                            Makefile.in REPORTING-BUGS
                                                                                                                                                                                                                                                    -
                                                            config.h.in
                                                                                                                                                                                                                                                    sopf nsm.h
#nsm_should_adj (nbr : struct osp config.log config.status finsm_timer_set (nbr : struct osp config.status finsm_twoway_received (nbr : work/quagga : work/quagga : /work/quagga : /work/q
                                                           config.log
                                                                                                                             mulitip6.sh
                                                                                                                                                                                                                                                    - ospf opaque c
                                                                                                                                                                                                                                                    ospf_opaque.h
                                                                                                                                                                                                                                                    ospf packet.c
                                                                                                                                                                                                                                                     e ospf packet.h
                                                                                                                                                                                                                                                    aspf route.c
                                                                                                                                                                                                                                                    - ospf route.h
                                                                                                                                                                                                                                                    aspf routemap.c
                                                                                                                                                                                                                                                    a ospf_snmp.c
                                                           /work/quagga # 1s
                                                                                                                                                                                 stamp-hl
                                                                                                                                                                                                                                                    sopf snmp.h
                                                                                         configure.ac
                                                                                                                                                                                                                                                    ospf_spf.c
                                                                                                                                                                                                                                                    a ospf spf.h
 sospf_db_summary_isempty :
                                                                                         COPYING LIB
                                                                                                                             ltmain.sh
                                                                                                                                                                                                                                                    aspf_te.c
 W-2 ospf nsm.c
                                                                                         escope.files
                                                                                                                                                                                                                                                    ospf te.h
   - option.c
                                                                                         cscope.out
                                                                                                                            Makefile
                                                                                                                                                     readMe.txt
                                                                                                                                                                                                                                                    espf vty.c
 # /work/fcms/cli/tunnel
                                                                                                                             Makefile.an
                                                                                                                                                                                                                                                    aspf vty.h
                                                                                                                                                                                                                                                    e ospf zebra.c
  - tunnel.c
                                                            config.h
                                                                                                                             Makefile.in
                                                                                                                                                                                                                                                    - ospf zebra.h
 ■ /work/fcms/cli/watchdog
   Licensestatus.c
                                                            config.log
                                                                                                                             mulitip6.sh
                                                                                                                                                                                                                                                    aspfd.c
 ■ /work/quagga/ospfd
                                                                                                                                                                                                                                                    ospfd.h
 m ...mis.os/rootdir/wsr/sbin
                                                            /work/quagga #
   □ L backpkg
                                                                                                  Bot of 5.9k (124,15) (EShell ht-p pair) -- 1:14PM 1.30 2011 -----
```

Emacs的特点

• 定制性

- ➤ Emacs用户可以根据自身的需要和偏好对编辑器进行定制。
- ➤ 定制Emacs主要有三种方法:

第一种方法是使用Emacs的交互式定制功能。

第二种方法是将一系列按键记录为宏,调用这一宏可以重复进行已经记录的一系列复杂操作。

第三种方法是通过使用Emasc Lisp完成Emacs的定制。

Emacs的特点 (续)

• 文档国际化

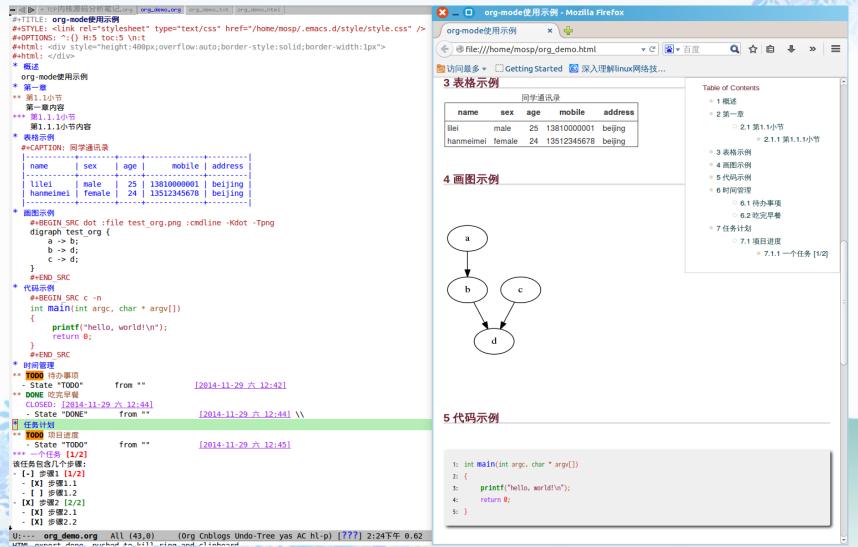
- ➤ Emacs支持对多种文字的文本编辑,包括UTF-8在内的诸 多编码系统,使得世界上大多数语言的用户都能通过 Emacs进行文本处理。
- ➤ Emacs还能通过调用诸如ispell这样的外部程序,实现多种语言的拼写检查。
- ➤ 在Emacs程序内部,每个字符都用1至4个字节表示,首字 节用于表示这一字符所属的编码系统。

文档编辑

- 写文档(org)
- 做幻灯片(beamer)
- 做表格
- 列编辑(cua)
- 记笔记(evernote)
- 写博客(cnblogs)
- 时间管理(gtd)
- 任务计划(gtd)
- · 思维导图(freemind)

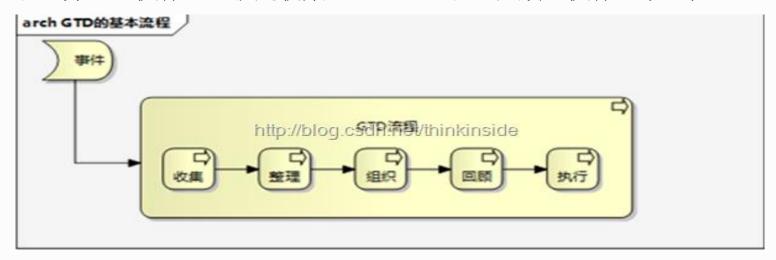
文档编辑---综合

Org-模式(Org-mode)是文本编辑软件Emacs的一种支持内容分级显示的编辑模式。这种模式支持写 to-do 列表,日志管理,做笔记,做工程计划或者写网页。



文档编辑(时间管理)

• GTD(Getting Things Done) 是一种将繁重超负荷的工作生活方式变成无压力高效的时间管理系统。其核心是把所有要做的事情都记录下来,然后整理组织,并逐一执行。也就是收集、整理、组织、回顾、执行五个过程。



收集: 把接收到的各种事件集中在一起;

整理:对事件做出快速判断,决定直接抛弃、留存归档或是产生一个任务;

组织:根据任务的不同属性将其归入不同的工作清单;回顾:定期检查工作清单,如果条件成熟就进行处理;

执行:对于可以处理的任务进行处理,并归档。

文档编辑(时间管理续)

如下图所示,左边为org为事件编辑,通过按下相应快捷键可以看当天/当月等已完成事情,未完成事件及提醒功能。

```
File Edit Options Buffers Tools Org Tbl Help
* 支持问题
                            Week-agenda (W52):
                            Monday 21 December 2009 W52
** DONE admin增加列表显示...
** TODO 新增模块演示版...
                            Tuesday 22 December 2009
                            Wednesday 23 December 2009
** TODO
            □盆问题...
                                          8:00.....
* wiki
                              work:
                                         9:00-10:00 Scheduled: DO
** TODO 自定义项与自由项
                                         10:00.....
* TODO 全文检索问题...
                                         12:00..... ---
 * DONE 电话会议,U盘加密问题...
                                         14:00..... Deadline:
                              home:
                                                                DO a
                                         14:00 . . . . . -
                                         16:00....
                                         18:00....
                                         20:00.....
   org文件区
                                         Scheduled: DONE admin増加列
                              work:
                              work:
                                         In 6 d.: TODO archeved
                                      24 December 2009
                            Thursday
                                                               新增
                              work:
                                         Scheduled: TODO
                              work:
                                        Scheduled: TODO
                                                              U盘F.
                                        Scheduled: TODO 确认一下转换
                              home:
                            Friday 25 December 2009
                                         Deadline: TODO 全文检索问题
                              work:
                                        Deadline: TODO 胡适口途自行
Agenda* Top (4,0) (Org
-U:--- work.org
```

文档编辑(思维导图)

如下图所示,思维导图(freemind)是一个很好地整理知识、表达思维的工具。 它长相比较像人脑神经,比较利于人的记忆。

```
#+TITLE: Org-mode
 ** 写文档
 ** 发布成html
 ** org-bable
U:--- mind.org
                                 (Org yas company) — 1月 6 17:04 1.42
                                 写文档。
   发布成html
                  Org-mode
                                 org-bable
                              7325,3016,33 (Image[png] yas company) —
    mind.png
                 All (1,0)
Type C-c C-c to view the image as text.
```

邮件、新闻组、浏览器

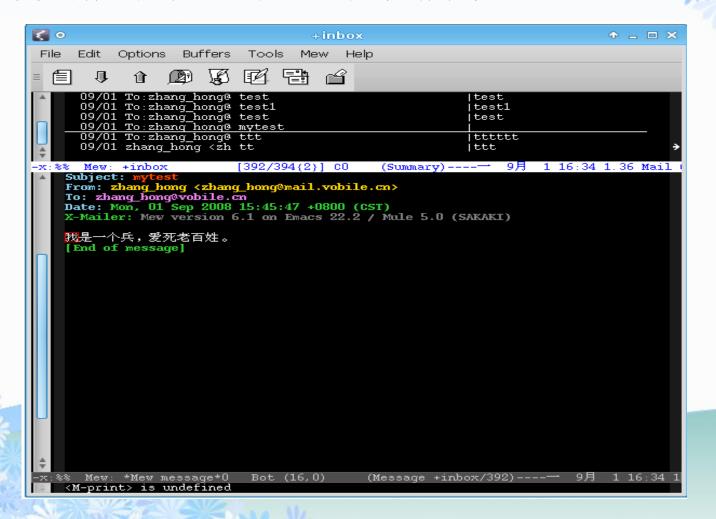
• 新闻阅读器(Gnus)

· 邮件管理器(Mew)

• 文本浏览器(w3m)

邮件(mew)、新闻组、浏览器

如下图所示,Mew是emacs里面一款很好很强大的邮件客户端插件, 把收发邮件也放到emacs中,可以提高工作效率。



邮件、新闻组、浏览器(w3m)

如下图所示,w3m为它的文本浏览器,使用原因有:

- 第一,平时在emacs里面再打开其他浏览器,不仅费时费力,而且当我编辑文档的时候看不见浏览器里面的内容,而当我看浏览器的时候又无法编辑文档。
- 第二,w3m是个文本浏览器,当我看到网页里面合适的内容,方便的复制 粘贴,这一点放在firefox或者谷歌的话,就得用鼠标一阵左右键了。
- 第三,这个是最重要的,生活在emacs里面是我的追求,尽管不乏偏执, ——但是这仍旧是一种追求,所以w3m必然不能缺少。。



文件、远程管理

• 文件管理器(Dired)

• 远程管理编辑(TRAMP)

文件(dired)、远程管理

如下图所示,dired 是 Emacs 自带的文件管理器,操作非常方便,再加上一些扩展之后无疑是一个理想的文件管理器。dired 最方便的一点就是可以对许多文件进行标记,并进行批量操作。查看某文件时也很方便。

```
~/work/fastweb-centos6.2/home/mosp/work/src/kernel/linux-2.6.32/net/ipv4/
     -rw-r--r-- 1 root root 41K 11月 15 22:10 ip gre.c
 -rw-r--r-- 1 root root 13K 11月 16 11:36 ip input.c
 -rw-r--r-- 1 root root 21K 11月 15 22:10 ipip.c
 -rw-r--r-- 1 root root 47K 11月 15 22:10 ipmr.c
 -rw-r--r-- 1 root root 15K 11月 16 11:36 ip options.c
 -rw-r--r-- 1 root root 36K 11月 16 11:36 ip output.c
 -rw-r--r-- 1 root root 29K 11月 16 11:36 ip sockglue.c
 -rw-r--r-- 1 root root 22K 11月 15 22:11 Kconfig
 -rw-r--r-- 1 root root 2.1K 11月 16 11:36 Makefile
* drwxr-xr-x 2 root root 4.0K 11月 16 11:41 netfilter
 -rw-r--r-- 1 root root 6.6K 11月 15 22:10 netfilter.c
 -rw-r--r-- 1 root root 17K 11月 16 11:36 proc.c
 -rw-r--r-- 1 root root 2.0K 11月 16 11:36 protocol.c
 -rw-r--r-- 1 root root 24K 11月 16 11:36 raw.c
 -rw-r--r-- 1 root root 85K 11月 16 11:36 route.c
 -rw-r--r-- 1 root root 10K 11月 15 22:10 syncookies.c
 -rw-r--r-- 1 root root 21K 11月 16 11:36 sysctl net ipv4.c
 -rw-r--r-- 1 root root 6.2K 11月 15 22:10 tcp bic.c
 -rw-r--r-- 1 root root 77K 11月 16 11:36 tcp.c
 -rw-r--r-- 1 root root 11K 11月 16 11:36 tcp cong.c
-rw-r--r-- 1 root root 14K 11月 16 11:36 tcp cubic.c
 -rw-r--r-- 1 root root 1.5K 11月 16 11:36 tcp diag.c
 -rw-r--r-- 1 root root 4.9K 11月 15 22:10 tcp highspeed.c
 -rw-r--r-- 1 root root 7.4K 11月 15 22:10 tcp htcp.c
 -rw-r--r-- 1 root root 4.9K 11月 15 22:10 tcp hybla.c
 -rw-r--r-- 1 root root 8.2K 11月 15 22:10 tcp illinois.c
 -rw-r--r-- 1 root root 165K 11月 16 11:36 tcp input.c
 -rw-r--r-- 1 root root 63K 11月 16 11:36 tcp ipv4.c
 -rw-r--r-- 1 root root 8.8K 11月 15 22:10 tcp lp.c
 -rw-r--r-- 1 root root 23K 11月 16 11:36 tcp minisocks.c
 -rw-r--r-- 1 root root 75K 11月 16 11:36 tcp output.c
 -rw-r--r-- 1 root root 6.0K 11月 15 22:10 tcp probe.c
 -rw-r--r-- 1 root root 1.5K 11月 15 22:10 tcp scalable.c
 -rw-r--r-- 1 root root 15K 11月 16 11:36 tcp timer.c
 -rw-r--r-- 1 root root 9.6K 11月 15 22:10 tcp vegas.c
                                      (Dired by name hl-p) 10:17下午 0.33
```

远程管理编辑(TRAMP)

• Emacs的TRAMP是用来访问scp/ssh/smb 服务器上的文件,可以做到像读写本地文 件一样编辑远程的文件。

• 通过TRAMP可以远程编辑或调试没有装 emacs环境的机器,并有代码着色功能,当 编辑完后保存时自动保存到远端。

翻译

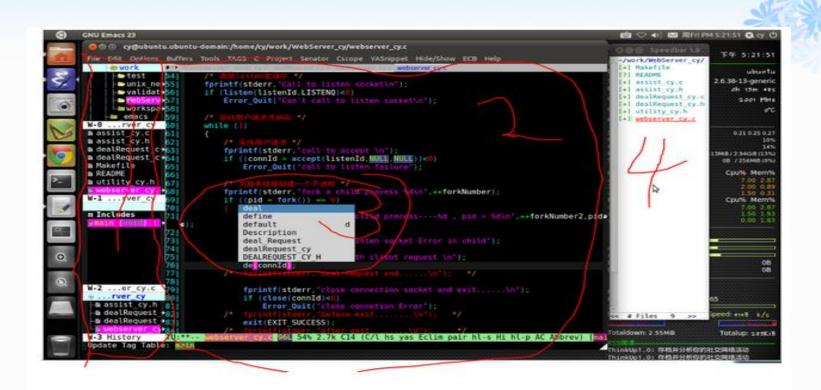
- 星际译王(stardict)
- 使用各种辞典

```
- S Meshellx Mbuffer-selections Mibuffers MOrg Export - Mo
<4/kernel/cpufreq $ kernel
                                                                                                                                                                                          ound 2 items, similar to kernel.
                                                                                                                                                                                           -->牛津英汉双解美化版
                                                                                                                                                                                           -->kernel
                                                                                                                                                                                          /'k3:nl; `k3:nl/ n
                                                                                                                                                                                          1 soft and usu edible part inside a nut or fr>
                                                                                                                                                                                          2 part of a grain or seed within the hard out>
                                                                                                                                                                                          3 (fig 比喻) central or essential part (of a s>
                                                                                                                                                                                          *the kernel of her argument 她的论据的核心.
                                                                                                                                                                                            -->21世纪英汉汉英双向词典
                                                                                                                                                                                          -->kernel
                                                                                                                                                                                          ker.nel
                                                                                                                                                                                          [`kʒnl; `kə:nl]
                                                                                                                                                                                             《源自古英语"子、果实"的意思》<<可数名词>>
                                                                                                                                                                                          1 (梅、桃等果实的)核,仁
                                                                                                                                                                                          2 (小麦等的) ?b粒
                                                                                                                                                                                          3 [问题等的]核心,重点,中心[of]
                                                                                                                                                                                          the ~ of a matter [question]
                                                                                                                                                                                           事件[问题]的核心
    U:--- *eshell*
                                                                                          All (1,67)
                                                                                                                                                   (EShell U U:**- *sdcv*
```

编程相关

- 适用多种语言编写
- 代码阅读(XCscope和ecb)
- 代码调试(gdb)
- 代码搜索定位(grep)
- 代码着色
- 自动补全
- 统一注释格式(doxymacs)
- 代码折叠
- 智能编译

编程相关(编码视图)



对照上图的编号:

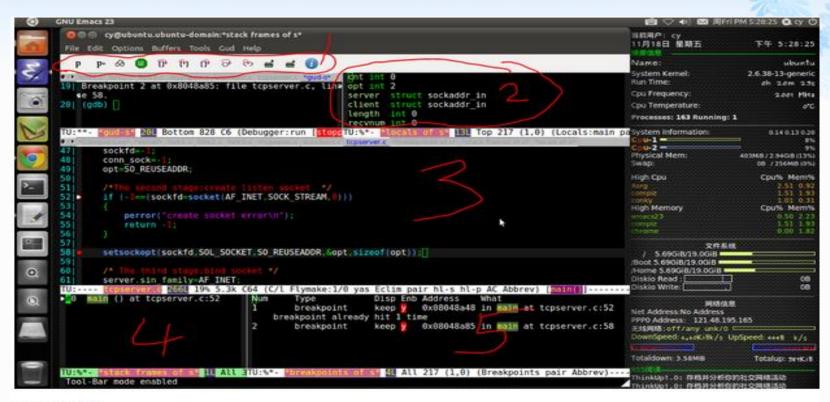
1:简洁明了的buffer浏览和操作,以及一些文件浏览器,tag浏览

2: 高亮代码编辑区

3:自动补全

4:更好的文件浏览器

编程相关(调试视图)



对照上图:

1:图形化调试选项

2:变量查看区

3:源码查看区

4: 栈查看区

5: 断点查看区

终端模拟器(eshell)

如下图所示,Eshell是Emacs完全用Elisp实现的类UNIX shell. 由于它完全是由Elisp实现的,因此它具有与Emacs相同的可移植性,而且它可以很自然的与Elisp代码相结合. 它跟emacs配合工作更加娴熟。

```
& _ • ...
- 🕒 🔊 *eshell* *buffer-selection* *Ibuffer* *Org Export/Publishing Help* | ipv4 | netfilter | *tramp/ssh2 root@192.168
~/study/linux/src/linux-2.6.11.12/arch/x86 64 $ pwd
/home/mosp/study/linux/src/linux-2.6.11.12/arch/x86 64
~/study/linux/src/linux-2.6.11.12/arch/x86 64 $ ls
boot defconfig ia32 Kconfig Kconfig.debug kernel lib Makefile mm oprofile pci
~/study/linux/src/linux-2.6.11.12/arch/x86_64 $ ll
总用量 72
drwx----- 4 mosp mosp 4096 1月 5 2004 boot
-rw-r--r-- 1 mosp mosp 22699 8月 26 2012 defconfig
drwx----- 2 mosp mosp 4096 1月 5 2004 ia32
-rw-r--r-- 1 mosp mosp 11507 8月 26 2012 Kconfig
-rw-r--r-- 1 mosp mosp 1961 8月 26 2012 Kconfig.debug
drwx----- 4 mosp mosp 4096 1月 5 2004 kernel
drwx----- 2 mosp mosp 4096 1月 5 2004 lib
-rw-r--r-- 1 mosp mosp 3886 8月 26 2012 Makefile
drwx----- 2 mosp mosp 4096 1月 5 2004 mm
drwx----- 2 mosp mosp 4096 1月 5
                                    2004 oprofile
drwx----- 2 mosp mosp 4096 1月 5 2004 pci
~/study/linux/src/linux-2.6.11.12/arch/x86 64 $
                                    (EShell Undo-Tree) 9:35下午 0.19
```

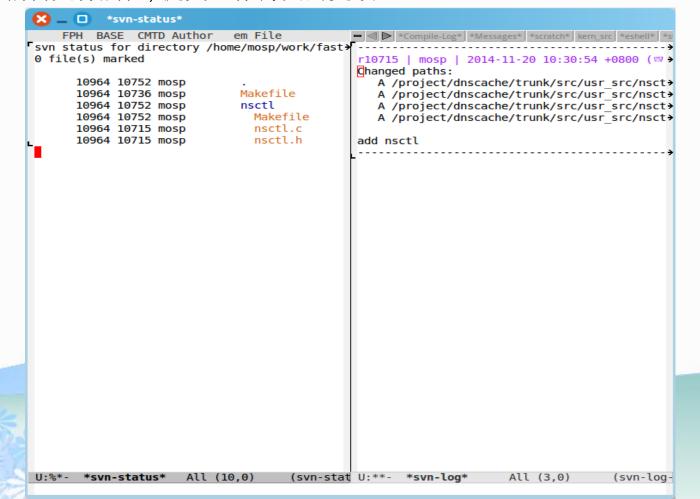
文件比较(ediff)

如下图所示,Emacs 提供了一个 diff 的接口,称为 Ediff ,将此工作变得简单。 不仅将 diff 的输出变得可读之外, Emacs 还提供了合并文件函数以及其他的应用 补丁,更多的细节信息,可以参考 Ediff 的 info 文档。 Emacs 还能比较不同文件夹 下的两个文件,若你在不同地方工作,这个功能可能会非常有用。

```
/home/mosp/study/linux/src/linux-2.6.11.12/net/new_compat.c
     org_demo.html new_compat.c compat.c
                                                                                                   ■ 🜒 🗩 org_demo.html | new_compat.c | compat.c
                                   char user *optual
                                                                           mouse-1: switch to buffer "compat.c"
                                                                          mouse-2: pop to buffer, mouse-3: delete other windows ure indep
          struct compat sock fproq
          struct sock fprog user *kfprog = compat*
          compat uptr t ptr;
                                                                                                    struct compat sock fproq {
          u16 len:
                                                                                                           u16
                                                                                                                           len:
                                                                                                           compat uptr t filter;
                                                                                                                                                                     /* struct soc>
          if (!access_ok(VERIFY READ, fprog32, size > );
                   !access_ok(VERIFY WRITE, kfprog, size>
                   get user(len, &fprog32->len) ||
                                                                                                  ▽static int do set attach filter(int fd, →
                   __get_user(ptr, &fprog32->filter) ||
                                                                                                                                    char user *optval, int optl>
                   put_user(len, &kfprog->len) ||
                      put user(compat ptr(ptr), &kfprog->>
                                                                                                           struct compat sock fprog __user *fprog32 >
                  return -EFAULT;
                                                                                                           struct sock_fprog __user *kfprog = compat>
                                                                                                           compat uptr t ptr;
          return sys setsockopt(fd, level, optname, >
                                                                                                           u16 len;
                                       sizeof(struct sock fprog));
                                                                                                           if (!access ok(VERIFY READ, fprog32, size>
                                                                                                                    !access_ok(VERIFY WRITE, kfprog, size>
▼Static int do set sock timeout(int fd, i)
                                                                                                                    get user(len, &fprog32->len) ||
                                                                                                                    get user(ptr, &fprog32->filter) ||
          struct compat timeval user *up = (struc*
                                                                                                                    __put_user(len, &kfprog->len) ||
                                                                                                                     put user(compat ptr(ptr), &kfprog->>
          struct timeval ktime;
          mm segment t old fs;
                                                                                                                    return -EFAULT;
          int err;
                                                                                                           return sys_setsockopt(fd, level, optname, >
          if (optlen < sizeof(*up))</pre>
                                                                                                                                        sizeof(struct sock fprog));
                  return -EINVAL;
          if (!access ok(VERIFY READ, up, sizeof(*u>
                  get_user(ktime.tv sec, &up->tv sec) >
                   get_user(ktime.tv_usec, &up->tv_use*\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\ri
                  return -EFAULT;
                                                                                                                                    char user *optval, int optl>
          old fs = qet fs():
          set fs(KERNEL DS);
                                                                                                           /* SO SET REPLACE seems to be the same in>
          err = sys setsockopt(fd, level, optname,
                                                                                                           if (optname == IPT SO SET REPLACE)
          set fs(old fs);
                                                                                                                    return do_netfilter_replace(fd, level>
                                                                                                                            new compat.c 71% (440.0)
                          compat.c
                                                          70% (440,0)
```

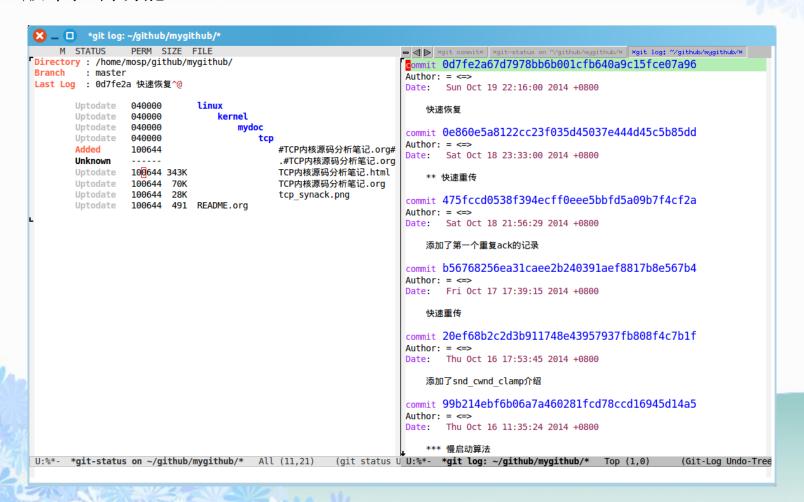
版本控制(psvn)

如下图所示,psvn.el 是 Emacs 的一个 Subversion 接口。它更方便的使用svn功能:显示Subversion工作拷贝的状态;显示不同的版本之间的差别;显示版本日志编辑文件属性;提交工作拷贝的更改。



版本控制(git)

如下图所示,在emacs用git功能,更加方便的进行git库的提交/查看等版本控制功能。



休闲娱乐

• 看电影听音乐(EMMS)

• 心理医生(doctor)

• 游戏(俄罗斯方块/贪食蛇)

• 屏幕保护(zone)

看电影听音乐

• 听音乐(EMMS)

如下图所示,EMMS 让你可以在 Emacs 里面播放多媒体文件,他被设计的小巧干净,使用外部播放器进行播放。是一个小巧、可扩展的完全由 Elisp 写成的多媒体系统,支持多种音频、视频格式以及流媒体播放(这事实上由你所使用的播放器决定),可以进行标签操作、方便的播放列表管理以及打分制度,总之,这正是我想要的。

ctivities	Sun 20:04		分 • 0 🖟 🖘 🖾 fl		
EMMS					
Current *eason* *westlife*					
. 愛是一本書	<认了吧>	陳奕迅	2007-04	04:24	
第一個雅皮士	<认了吧>	陳奕迅	2007-04	03:24	
0. 白色球鞋 1. 瑪利奥派對 2. 熱島小夜曲	< 认了吧> [<listen cd1="" chan="" eason="" to=""> <listen cd1="" chan="" eason="" to=""></listen></listen>	陳奕迅 陳奕迅 陳奕迅	2007-04 2007-10 2007-10	04:38	
				03:26	
				04:05	
. 閃	<listen cd1="" chan="" eason="" to=""></listen>	陳奕迅	2007-10	03:42	
,演唱會	<listen cd1="" chan="" eason="" to=""></listen>	陳奕迅	2007-10	02:36	
. 滑鐵盧車站	<listen cd1="" chan="" eason="" to=""></listen>	陳奕迅	2007-10	03:48	
Crying in the Party	<listen cd1="" chan="" eason="" to=""></listen>	陳奕迅	2007-10	04:13	
狂熱革命	<listen cd1="" chan="" eason="" to=""></listen>	陳奕迅	2007-10	03:44	
變色龍	<listen cd1="" chan="" eason="" to=""></listen>	陳奕迅	2007-10	03:46	
時代巨輪	<listen cd1="" chan="" eason="" to=""></listen>	陳奕迅	2007-10	03:38	
·乜嘢啫 ·瑪利奥派對	<pre><listen cd1="" chan="" eason="" to=""> <listen cd2="" chan="" eason="" to=""></listen></listen></listen></listen></listen></listen></listen></listen></pre>	陳奕迅 陳奕迅 陳奕迅 陳奕迅 陳奕迅 陳奕迅	2007-10 2007-10 2007-10 2007-10 2007-10 2007-10 2007-10	03:39	
				06:03	
熱島小夜曲				05:47	
				05:54	
演唱會				04:25	
. 滑鐵盧車站					
Crying in the Party					
狂熱革命	<listen cd2="" chan="" eason="" to=""></listen>	陳奕迅	2007-10	04:42	
變色龍	<listen cd2="" chan="" eason="" to=""></listen>	陳奕迅	2007-10		
時代巨輪	<listen cd2="" chan="" eason="" to=""> <listen cd2="" chan="" eason="" to=""></listen></listen>	陳 奕 迅 陳奕迅	2007-10 2007-10	05:44 03:58	
也嘢啫					
重口味		陳奕迅		03:59	
非禮		陳奕迅		03:13	
3.Class	<3mm> <3mm>	陳奕迅 陳奕迅		03:13	
碌卡				03:37	
笑死 朕		陳奕迅		03:20	
蚊		陳奕迅		03:45	
Let It Out		陳奕迅		04:04	
3.習慣說		陳奕迅		03:48	
信任		陳奕迅		03:31	
%*- *eason* 92% (395,0)	(EMMS) り [08.天一半, 地一半	<新。	生活(新曲,精选)>	F)	陳奕.

看电影听音乐

• 看电影(EMMS)



心理医生

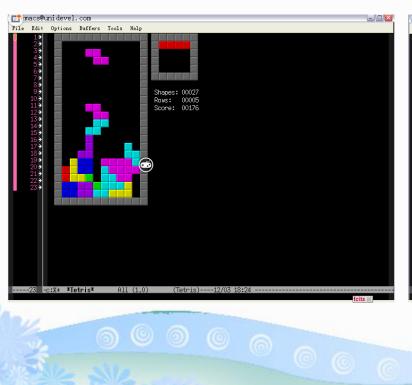
• 心理医生(doctor)

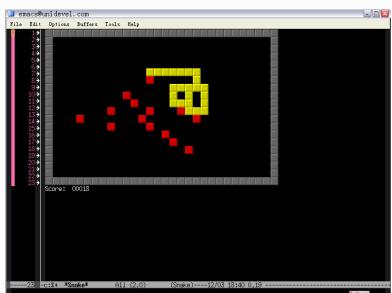
doctor是emacs内置的心理医生,当你无聊或者烦躁的时候可以跟它聊聊,也许让自己更加舒坦一些或者压力得到释放,它只支持英文,你也可以当成练习英语交

```
I am the psychotherapist. Please, describe your problems. Each time you are
finished talking, type RET twice.
hello
How do you do? What brings you to see me?
I am sleepy
Why do you say you are sleepy?
Because I study erveryday
Is it because you study erveryday that you came to me?
maybe
Can you elaborate on that?
no
Because I not have a lot of money
Does it bother you that you not have a lot of money?
因为北京的房价太高了,都10万每平,还有没有天理
Earlier you said you study erveryday?
I see... Well, what makes you believe this is so?
I am tired
U:**- *doctor*
```

游戏

• 游戏(俄罗斯方块或贪食蛇等)





聊天通讯

• IRC聊天

IRC

本文示例,默认连接的 IRC 服务器为 : irc.freenode.net

加入 channel (房间)

当我们连接到一个 IRC 服务器后,就可以使用 "/join" 命令加入感兴趣的房间聊天。

/join #ubuntu-cn /join #fedora-cn

聊天

使用 IRC 不聊天做什么?任何客户端下,只要连接到一个 IRC 服务器 (最用的 是 irc.freenode.net),加入 channel (或者称呼为房间),就可以说话了。

指定用户名发信息

如果信息前想加入某个人的名字,这样能让别人注意到你的信息,却又要大家都能看见。可以先输入该用户名(用 Tab 键补全),再输入信息即可:<Jian> jianlee: 这是一条测试信息。

私聊

为了不打扰大家,私聊是一种礼貌。

使用 "/msg 用户名"

/msg nickname "信息"

这样就可以和 nickname 私聊了。

使用 "/query 用户名"

如果要和 "Jian" 私聊,可以这样

/query Jian

设置自己的 nickname

注册 nickname

做美食

• 煮咖啡、炒川菜

RFC中支持超文本咖啡壶协议(RFC2324)与茶壶协议(RFC7168),而emacs提供了完全支持RFC2323的coffee.el插件提供控制展示界面,只要你有一根网线,加一个支持此协议的咖啡壶,你确实可以用emacs煮咖啡,甚至有人已经用它来炒过川菜。







Lisp语言介绍

LISP语言被称之为表处理语言,它的基本数据结构就是表。 LISP语言在人工智能历史上有着重要的地位,80年代中期以前的人 工智能系统几乎都是用LISP语言实现的。

LISP是一种计算机的表处理语言,它是英文List Processing的缩写。自从1960年J. McCathy在美国MIT首先发表LISP以来,很快就被人工智能工作者所接受,成为在人工智能领域应用的最广泛的计算机语言。人工智能所取得的各项成果,都是和LISP的功劳分不开的,人们曾这样对LISP语言做出评价: LISP语言是人工智能的数学,不仅对人工智能的机器实现有重要意义,而且是人工智能理论研究的重要工具。LISP语言武装了一代人工智能科学家。

Lisp语言介绍(二)

这是一个很实用的功能,实际上也是我的一个主要目的,这么强大的语言,用好了,比shell脚本方便多了,而且性能也搞。甚至可以代替Java编写程序。 现在我的test.el文件内容如下:

```
[plain] view plain copy print ?

01. (defun add2 (x)

02. (+ 2 x))

03. 
04. (print (add2 8))
```

然后执行下面的命令:

```
[plain] view plain copy print ?

01. emacs --no-site-file --script /home/chenshu/test.el
```

结果显示10

elisp也可以运行外部程序,比如

```
[plain] view plain copy print ?

01. (start-process "shell" nil "curl" "http://www.baidu.com")
```

创建一个shell进程,里面运行了curl命令发出http请求。

这是一个简单的练习,如果一个list中有几个数字,写一个函数,找出最大值。

```
[plain] view plain copy print ?

01. (setq number-list '(5 8 9 10 87 34))

02.

03. (defun max-number(n-list)

04. (let ((tmp (car n-list)) (max tmp))

05. (while n-list)

06. (setq tmp (car n-list))

07. (if (> tmp max)

08. (setq max tmp)

09. )

10. (setq n-list (cdr n-list))

11. )

12. max

13. )

14. )

15. [max-number number-list)
```

Lisp语言介绍(三)

• 优点

如果我们把流行的编程语言,以这样的顺序排列: Java、Perl、Python、Ruby。你会发现,排在越后面的语言,越像Lisp。

Python模仿Lisp,甚至把许多Lisp黑客认为属于设计错误的功能,也一起模仿了。至于Ruby,如果回到1975年,你声称它是一种Lisp方言,没有人会反对。 编程语言现在的发展,不过刚刚赶上1958年Lisp语言的水平。

1958年,John McCarthy设计了Lisp语言。我认为,当前最新潮的编程语言,只是实现了他在1958年的设想而已。

这怎么可能呢? 计算机技术的发展,不是日新月异吗? **1958**年的技术,怎么可能超过今天的水平呢?

让我告诉你原因。

这是因为John McCarthy本来没打算把Lisp设计成编程语言,至少不是我们现在意义上的编程语言。他的原意只是想做一种理论演算,用更简洁的方式定义图灵机。

所以,为什么上个世纪50年代的编程语言,到现在还没有过时?简单说,因为这种语言本质上不是一种技术,而是数学。数学是不会过时的。你不应该把Lisp语言与50年代的硬件联系在一起,而是应该把它与快速排序(Quicksort)算法进行类比。这种算法是1960年提出的,至今仍然是最快的通用排序方法。

参考: http://www.ruanyifeng.com/blog/2010/10/why_lisp_is_superior.html 《黑客与画家》中的[为什么Lisp语言如此先进?]。

Lisp语言介绍(四)

• 优点

学习Lisp所收获的是如何"自由地"表达你的思想,这正是Lisp最大的魅力所在,也是这门古老的语言仍然具有很强的生命力的根本原因。

参考: http://coolshell.cn/articles/7526.html

《Lisp的永恒之道》

Lisp使人超越平庸,走到前沿。学会Lisp意味着你能找到更好的工作,因为聪明的雇主会被你与众不同的洞察力所打动。学会Lisp也可能意味着明天你可能会被解雇,因为你总是强调,如果公司所有软件都用Lisp写,公司将会如何卓越,而这些话你的同事会听烦的。Lisp值得努力学习吗?那些已经学会Lisp的人都说值得,当然,这取决于你的判断。

参考: http://www.csdn.net/article/2012-11-22/2812113-The-Nature-Of-Lisp《Lisp的本质》

Emacs vs vi

- vi可以快速在文档中移动和修改
- 难入入门而功能有限,对于初学这来说不知道从何开始
- 两种模式的切换: 命令模式, 编辑模式。
- vi的编写者尝试使它能通过300波特率的调制解调器使用
 - ,所以这款编辑器被优化成当屏幕显示极慢时也能用
- 组合特定动作和移动,提供高级方式

Emacs vs vi (二)

- Emacs支持无穷的定制性
- 功能丰富,不仅是文本编辑器

- 完备的工作环境,可以通过编写Lisp代码来添加新功能
- 提供多模式,每个模式有唯一的命令集,不论要实现什么功能,切换到相应模式就好了

Emacs vs vi (三)

- vi和Emacs两大编辑器冷战不断
- 两者的区别主要在于灵活性和功能性
- 两者都提供一些高级方式来减少键盘上的手指速度和编程 者大脑速度的差异
- vi需要耐性去学习,在任何Unix系统上都获得了行动自由
- Emacs赋予你疯狂的自由来自定义合适的装备,但如果不 小心,会成为自己配置的囚徒。

Emacs缺点

- 单线程
- 配置复杂
- 启动慢
- 不易上手
- · 容易患上 Emacs "综合症", 上瘾, 在没有Emacs的电脑前感到痛苦,如果哪一天不用emacs你会觉得世界暗了下来,

了无情趣。

