

# Conky 中文文档 v2.0

制作：wtry  
原著：canghaiwuhen

2010-7-14

# 安装

一些常用 linux 版本的安装方法：

gentoo                   -> emerge app-admin/conky

debian/ubuntu -> apt-get install conky 或者 aptitude install conky

archlinux   -> pacman -S conky 或者 yaourt -S conky

编译安装方法为：

下载源码解压 然后进入解压目录

```
sh autogen.sh
./configure --prefix=/usr --mandir=/usr/share/man
--infodir=/usr/share/info --datadir=/usr/share --sysconfdir=/etc
--localstatedir=/var/lib --disable-own-window --enable-audacious[=yes|no|legacy]
--enable-bmpx --disable-hddtemp --disable-mpd --enable-xmms2 --disable-portmon
--disable-network --enable-debug --disable-x11 --disable-double-buffer
--disable-xdamage --disable-xft

make
make install
src/conky
```

# 选项

命令行下启动 conky 时可用的选项

conky [options]

-v | -V | --version

显示版本并退出

-q | --quiet

quiet 模式运行 conky 也就是无输出

-D | --debug

debug 模式运行 conky 会显示更详细的输出

-a | --alignment=ALIGNMENT

自定义 conky 的显示位置 {top,bottom,middle}\_{left,right,middle} 比如 top\_right.

-b | --double-buffer

双缓冲模式(消除闪烁)

-c | --config=FILE

自定义加载配置文件

-C | --print-config

显示默认配置文件 -v 参数显示的信息中有该文件位置

-d | --daemonize

后台保护 conky 进程

-f | --font=FONT

自定义使用的字体

-h | --help

显示帮助并退出

-o | --own-window  
在自己窗口运行 conky

-t | --text=TEXT  
调用临时 TEXT 用单引号引起来 比如 -t '\$uptime'

-u | --interval=SECONDS  
自定义刷新时间

-w | --window-id=WIN\_ID  
定义窗口 ID

-X | --display=DISPLAY  
使用 X11 显示

-x X\_COORDINATE  
X 轴位置

-y Y\_COORDINATE  
Y 轴位置

-i COUNT  
定义刷新次数 完成该次数后退出

关闭 conky 用的命令为 `killall conky`

重启 conky 用的命令为 `killall -SIGUSR1 conky`

## 配置

配置文件位于 `$HOME/.conkyrc` 或者 `${sysconfdir}/conky/conky.conf`

示例文件 `/etc/conky/conky.conf`

在 <http://conky.sf.net/> 可以找到很多配置文件

TEXT	启动以后显示比如主机名 内核 用户那些所用的文本 也就是配置文件的主体部分
alias	创建变量 第一个参数是新名称 第二个是老名称 其他的参数都是传递给变量的 比如 以 \$yun 代替 \${yun si feng} 应写成 <code>alias yun yun si feng</code> 注意 这仅仅是创建一个别名你也可以用环境变量 在开始的时候写明 <code>yun="yun si feng"</code>
alignment	屏幕中的位置 可以是 <code>top_left</code> , <code>top_right</code> , <code>top_middle</code> , <code>bottom_left</code> , <code>bottom_right</code> , <code>bottom_middle</code> , <code>middle_left</code> , <code>middle_right</code> 也可以缩写为 <code>tl</code> , <code>tr</code> , <code>tm</code> , <code>bl</code> , <code>br</code> , <code>bm</code> , <code>ml</code> , <code>mr</code> 也可以不设置 直接看 <code>gap_x</code> 和 <code>gap_y</code>
append_file	附加文件作为注释
background	是否嵌入背景 这是布尔值 真则为嵌入 假则不嵌入
border_margin	边框周边的空白 以像素为单位
border_width	边框线的宽度 以像素为单位
colorN	预先定义 TEXT 中使用的颜色 N 是从 0 到 9 的数字 当以颜色以十六进制表示的时候可以省略开头的 #
cpu_avg_samples	cpu 数据监测模式 一般选 2
default_bar_size	默认进度条的宽度和高度 例如: <code>default_bar_size 0 6</code> 注意前面的是宽度 这对于没有具体大小参数的执行程序的进度条( <code>execbar</code> 和 <code>execibar</code> )很有用
default_color	默认颜色

default_gauge_size	仪表显示用的仪表(样式类似汽车的仪表盘)的宽度和高度例如：
default_gauge_size 25 25	
	这对于没有具体大小参数的执行程序的显示仪表(execgauge和execigauge)很有用
default_graph_size	频谱显示用的框图的宽度和高度 例如：default_graph_size 0 25
	这对于没有具体大小参数的执行程序的监测表(execgraph和execigraph)很有用
default_outline_color	默认边框外缘的颜色
default_shade_color	默认阴影颜色和边界的阴影颜色
diskio_avg_samples	硬盘输入输出监测
display	X窗口的连接
double_buffer	消除闪烁 需要 own_window 支持
draw_borders	创建字体边框
draw_graph_borders	创建图形边框
draw_outline	创建边缘
draw_shades	创建阴影
font	字体设置 可以用 xfontsel 选择你喜欢的
gap_x	与屏幕左右边界的距离 以像素为单位
gap_y	与屏幕上下边界的距离 以像素为单位
if_up_strictness	监视端口 比如 上传 连接 分配的 ip 地址
imap	默认 IMAP 服务器参数为 :host user pass [-i interval (in seconds)] [-f folder] [-p port] [-e command] [-r retries] 默认端口 143 默认文件夹 INBOX 默认时间间隔 5 分钟 默认重试次数 5 如果设置了密码 启动 conky 时会提示输入密码
imlib_cache_size	MLIB2 图像缓存大小 单位为字节 默认 用 \$image 设置参数设置为 0 禁用图像缓存
lua_load	加载 lua 脚本 用空格分隔
mail_spool	邮件检查
max_port_monitor_connections	监视的最大端口连接数 默认 256
max_specials	特殊选项的最大数 比如字体 偏移 默认 512
max_user_text bytes	用户文本缓冲区的大小 也就是 TEXT 部分的大小 默认 16384 字节
maximum_width(height) pixels	窗口最大宽度(高度) 单位为像素
minimum_size width(height)	窗口最小宽度(高度) 单位为像素
mpd_host	MPD 服务器地址
mpd_password	MPD 服务器密码
mpd_port	MPD 服务器端口
music_player_interval	音乐播放器线程刷新时间间隔 (默认为 Conky 的刷新间隔)
net_avg_samples	网络传输平均数据监测
no_buffers	从内存中清除文件系统缓冲区
out_to_console	输出文本到标准输入 也就是终端
out_to_stderr	输出文本到错误输入
out_to_x	设置为 no 将不会有任何输出在 X(和 out_to_console 差

不多) 如果设置为 no 必须确认这是位于所有其他的 X 相关的设置之前  
(以第一条的 configfile 为准) 默认 yes

override_utf8_locale	强制 uft8 编码需要 xft 支持
overwrite_file	覆写文件作为注释
own_window	布尔值 创建自主窗口
own_window_class	手动设置 WM_CLASS 名称 默认为 Conky
own_window_colour	如果窗口不透明 则需设置颜色 默认为黑 参数为十六进制颜色 或者 RGB 颜色名字(/usr/share/X11/rgb.txt)
own_window_hints	参数为 undecorated,below,above,sticky,skip_taskbar
skip_pager	如果 own_windows 为 yes 就能用这个得到 conky 窗口提示 注意 如果用了 own_window_type 窗口提示就会被忽略
own_window_title	手动设置窗口的名称 默认为<hostname> - conky
own_window_transparent	布尔值 窗口是否透明
own_window_type	如果 own_window 为 yes 可以指定窗口样式为 normal, desktop, dock 或者 override 默认为 nomal desktop 是指没有装饰置顶 不显示在 pager 和 taskbar 上 出现在所有 workspace 上的的特殊窗口 override 指的是不在控制之下的窗口 会忽略提示 某些时候很有用
pad_percents	小数的百分比形式 0 没有百分比形式
pop3	默认 POP3 服务器参数为 host user pass [-i interval (in seconds)] [-p port] [-e command] [-r retries] 默认端口 110 默认时间间隔 5 分钟 默认重复次数 5 次 如果没有密码 启动 conky 的时候会提示输入
short_units	缩写单位字符 比如 kib->k Gib->G 默认为 off
show_graph_range	频谱图的时间范围
show_graph_scale	频谱图显示最大数值
stippled_borders	虚线框 单位像素
temperature_unit	期望输出电脑的所有器件显示温度 参数是 fahrenheit(华氏) 或 celsius(摄氏)默认的单位是摄氏度
templateN	为后面 TEXT 中的使用定义一个模板 N 为 0 到 9 的数字 TEXT 中的一些转义: '\n' -> 换行 '\' ' -> 下划线 '\ ' -> 空格 '\N' -> 模板参数 N
text_buffer_size bytes	标准文本缓冲区的大小 默认 256 字节 增大缓冲区会降低 conky 的性能但会增加显示的文本的数量 设置的大小不能小于默认的 256 字节
top_cpu_separate	设为 true CPU 的上方显示一个在使用处理器的功率 设为 false CPU 的上方显示在使用的所有处理器的功率总和
top_name_width	顶端名字\$top 的宽度 默认 15 字节
total_run_times	conky 运行的次数 指刷新的次数 0 为永远运行
update_interval	刷新的间隔时间 单位为秒 间隔时间越长 系统负担越轻 监控实时性越差

uppercase	大写变换 设置为 true 输出的所有字母变大写
use_spacer	增加部件周围的空间 阻止它们移动到别的周边 参数有 left, right 和 none(默认的) 注意 这只是使用等宽字体是需要的 比如 Bitstream Vera Sans Mono
use_xft	使用 xft 反锯齿(anti-aliased font 和 stuff)
xftalpha	xft 字体的开头 必须是一个数值或者 0 和 1
xftfont	使用 xft 字体

## 函数

颜色一般用的是十六进制参数可以看 /usr/share/X11/rgb.txt

也可以直接对照 <http://sedition.com/perl/rgb.html>

acpiacadapter	使用 ACPI 的话显示的 AC 适配器
acpifan	使用 ACPI 的话显示风扇状态
acpitemp	使用 ACPI 的话显示 cpu 的摄氏温度
addr interface	IP 地址 如果没分配 则 "No Address"
addrs interface	IP 地址(如果分配了地址就合 addr 一样) 仅在 linux 下有效
adt746xcpu	CPU 温度 得自 therm_adt746x
adt746xfan	风扇速度 得自 therm_adt746x
alignc (num)	文本中心对齐
alignr (num)	文本右对齐
apcupsd host port	建立连接 apcupsd 的服务 无输出 默认本地端口 3551
apcupsd_cable	输出 UPS 的连接类型
apcupsd_charge	当前电池电量的百分比
apcupsd_lastxfer	线性表示电池的剩余电量
apcupsd_linev	额定输入电压
apcupsd_load	电流负载的百分比
apcupsd_loadbar	电流负载的进度条
apcupsd_loadgauge (height),(width)	电流负载仪表图
apcupsd_loadgraph	电流负载频谱图
apcupsd_model	输出 UPS 模式
apcupsd_name	输出 UPS 用户自定义的名称
apcupsd_status	线性输出电池当前状态
apcupsd_temp	当前内部温度
apcupsd_timeleft	电池支撑的剩余时间
apcupsd_upsmode	输出 UPS 的模式(比如独立的)
apm_adapter	显示 APM AC 适配器状态(只适用 FreeBSD)
apm_battery_life	百分比显示 APM 电池剩余电量(只适用 FreeBSD)
apm_battery_time	以 hh:mm:ss 或者 unknown 显示 APM 剩余电量(只适用 FreeBSD)
audacious_bar (height),(width)	进度条
audacious_bitrate	当前比特率
audacious_channels	声道
audacious_filename	完整路径名和文件名

audacious_frequency	采样率
audacious_length	总长度 格式 mm:ss
audacious_length_seconds	总长度 以 s 表示
audacious_main_volume	主音量
audacious_playlist_length	播放列表长度 即列表歌曲数目
audacious_playlist_position	当前播放的歌曲在列表中的位置
audacious_position	歌曲在播放时间的位置 格式 mm:ss
audacious_position_seconds	歌曲在播放时间的位置 以 s 表示
audacious_status	播放状态(Playing/Paused/Stopped/Not running)
audacious_title (max length)	歌曲标题的最大长度
battery (num)	百分比表示 ACPI 或者 APM 电池状态和剩余电量 参数为 ACPI 电池数量 默认为 BAT0
battery_bar (height),(width) (num)	电池剩余电量进度条 参数为 ACPI 电池数量默认 BAT0
battery_percent (num)	电池剩余电量百分比 参数为 ACPI 电池数量默认 BAT0
battery_short (num)	batter 的缩简模式
battery_time (num)	电池充放电剩余时间 默认参数 BAT0
bmpx_album	BMP 播放器的专辑
bmpx_artist	BMP 播放器的艺术家
bmpx_bitrate	BMP 播放器的比特率
bmpx_title	BMP 播放器的标题
bmpx_track	BMP 播放器的播放曲目的位置
bmpx_uri	BMP 播放器连接的网址
buffers	内存使用总量
cached	内存剩余总量
color (color)	改变绘图用的颜色为参数设置的颜色
colorN	改变绘图用的颜色参数 参数 0-9 之间
combine var1 var2	连接两个变量 比如 \${combine \${head /proc/cpuinfo 2} - \${head /proc/meminfo 1}} 输出为 第一行"cpuinfo_line1 - meminfo_line1" 第二 行"cpuinfo_line2 -"
conky_build_arch	conky 的 cpu 构架
conky_build_date	conky 的日期
conky_version	conky 的版本
cpu (cpuN)	cpu 使用率 cpu 的标号作为参数
cpubar (cpuN) (height),(width)	cpu 使用率的进度条
cpugauge (cpuN) (height),(width)	cpu 使用仪表图
cpugraph (cpuN) ("normal" "log") (height),(width) (gradient colour 1) (gradient colour 2) (scale)	cpu 使用频谱图 如果选 log 则是用对数标度两个颜色是颜色 渐变参数 用十六进制表示的话 可以去掉# scale 是刻度参数
disk_protect device	磁盘保护状态 如果支持的话输出 frozen 或 free
diskio (device)	显示当前的磁盘分区读写速度设备可选比如 sda 参数

为/dev/sda 也可以选择单独的分区

diskio_read (device)	显示磁盘读取速度
diskio_write (device)	显示磁盘写入速度
diskiograph (device) ("normal" "log") (height),(width) (gradient colour 1) (gradient colour 2) (scale)	磁盘读写频谱图 参数设置与 cpugraph 相似
diskiograph_read (device) ("normal" "log") (height),(width) (gradient colour 1) (gradient colour 2) (scale)	磁盘读取频谱图 参数设置与 cpugraph 相似
diskiograph_write (device) ("normal" "log") (height),(width) (gradient colour 1) (gradient colour 2) (scale)	磁盘写入频谱图 参数设置与 cpugraph 相似
downspeed net	下载速度 单位 KiB
downspeedf net	下载速度 单位在 KiB 精确到小数位
downspeedgraph (netdev) ("normal" "log") (height),(width) (gradient colour 1) (gradient colour 2) (scale)	下载速度频谱图 参数设置与 cpugraph 相似
draft_mails (maildir) (interval)	通过在配置文件中加入相关参数可以实现对电子邮箱的监视，每隔一段时间会收取服务器上的邮件，Conky 会监视系统中的 mail 管道并显示未阅读的邮件的数量邮箱类型为 draft
else	如果前面的为假 则~~
endif	\$if 的结束语句
entropy_avail	加密情况
entropy_bar (height),(width)	entropy_avail 的进度条
entropy_poolsize	密钥的体大小
eval string	字符串解析 根据 TEXT 对定义的变量字符串进行解析
eve api_userid api_key character_id	从 API 服务器获取 EVE 网络游戏账号情况
exec command	执行一个 shell 命令并在 conky 显示 警告：这会花费很多资源 建议写个 c 程序或者打个补丁
execbar command	和 exec 相似 但当第一个变量返回值为 0-100 时使用
该标号的进	度条的大小通过 default_bar_size 配置
execgauge command	和 execbar 相似只是把进度条换成仪表图 通过
	default_graph_size 配置
execgraph ("normal" "log") (height),(width) (gradient colour 1) (gradient colour 2) (scale)	使用频谱图的 execbar 参数和 cpugraph 相似 尺寸大小由
command	default_graph_size 配置
execi interval command	和 exec 相似只是间隔时间不同 不能少于
	update_interval 的配置 另见 \$texeci
execibar interval command	间隔时间不同的 execbar
execigauge interval command	间隔时间不同的 execgauge
execigraph interval command	间隔时间不同的 execgraph
execp command	类似 exec 但可以插入类似 \${color red}hi!\${color} 的脚本 警告：execp 每一次 conky 循环会销毁所有对象 比如用 \$execi 在 \$execp 建立一个声明 它将运行在功能相同但时间间隔为 execp 的声明运行
execpi interval command	时间间隔不同的 execp
flagged_mails (maildir) (interval)	类似 draft_mails 邮箱类型为 flagged
font (font)	字体设置 用于当前和后面的 可以用 \$font 而不需要参数 改回预设字体 和 \$color 类似
forwarded_mails (maildir) (interval)	类似 draft_mails 邮箱类型为 forwarded
freq (n)	频率 返回 cpu 的频率值单位 MHz 参数为 cpu 标号 默认为 1
freq_g (n)	单位为 GHz 的 freq



<code>fs_bar (height),(width) fs</code>	进度条表示文件系统使用的空间
<code>fs_bar_free (height),(width) fs</code>	进度条表示文件系统剩余的空间
<code>fs_free (fs)</code>	文件系统剩余空间
<code>fs_free_perc (fs)</code>	文件系统剩余空间百分比
<code>fs_size (fs)</code>	文件系统空间总大小
<code>fs_type (fs)</code>	文件系统类型
<code>fs_used (fs)</code>	文件系统使用空间
<code>fs_used_perc (fs)</code>	文件系统使用空间百分比
<code>goto x</code>	下一个元素输出到位置 x
<code>gw_iface</code>	显示默认路由的接口 或者"multiple" "none"(太多 没有)
<code>gw_ip</code>	显示默认网关的 IP 或者"multiple" "none"
<code>hddtemp dev, (host,(port))</code>	显示 hddtemp 服务运行的主机和端口并以报告形式输出指定硬盘的温度 预设主机 127.0.0.1 默认端口 7634
<code>head logfile lines (interval)</code>	显示前 N 行日志文件的文本如果没设置时间间隔 默认为 2 倍 conky 时间间隔 在文本缓冲区足够的情况下最多显示 30 行
<code>hr (height)</code>	水平实线 用来分隔区块 单位像素
<code>hwmon (dev) type n (factor offset)</code>	2.6 内核中 sysfs 的 hwmon 传感器 如果只有一个 hwmon 器件可以省略参数 参数中 in 和 vol 指的是电压 fan 指风扇 temp 指温度 参数 n 指的传感器标号 可以查看 /sys/class/hwmon/ 最后的可选参数 factor offset 允许 raw 输入 运算式为 $input = input * factor + offset$ 注意必须为十进制也就是说至少包含一个小数位
<code>i (dev) type n (factor offset)</code>	2.6 内核中 sysfs 的 IC2 总线传感器可以查看 /sys/bus/i/devices/ 参数类似 hwmon
<code>iconv_start codeset_from codeset_to</code>	GNU iconv 编码转换 用 iconv_stop 停止
<code>iconv_stop</code>	停止 iconv 编码转换
<code>if_empty (var)</code>	如果 conky 变量 var 为空 显示所有 \$if_empty 到与之匹配的 \$endif 之间内容
<code>if_existing file (string)</code>	如果 file 存在显示所有 \$if_existing 到与之匹配的 \$endif 之间内容可选的第二参数用来检查文件是否包含指定字符串
<code>if_gw</code>	如果至少有一个默认网关 显示所有 \$if_gw 到与之匹配的 \$endif 之间内容
<code>if_match expression</code>	布尔式判断 是否显示所有 \$if_match 到与之匹配的 \$endif 之间内容取决于判断结果的真假 正确形式包括左右式和判断符左右式的类型有： double: 双精度 有小数位 long: 长整型 整数 string: 字符串 单引号或者双引号中间的 double 和 long 判断符包括: '>', '<', '>=', '<=', '==' 和 '!='
<code>if_mixer_mute (mixer)</code>	如果有混音器 显示所有 \$if_mixer_mute 到与之匹配的 \$endif 之间内容 如果没有 则使用 master 音量
<code>if_mounted (mountpoint)</code>	如果挂载点挂载了硬件 显示所有 \$if_mounted 到与之匹配的 \$endif 之间内容
<code>if_mpd_playing</code>	如果 mpd 正在播放或者暂停显示所有 \$if_mpd_playing 到与之匹配的 \$endif 之间内容
<code>if_running (process)</code>	如果 process 进程正在运行显示所有 \$if_running 到与之匹配的 \$endif 之间内容这里使用 ``pidof`` 命令 支持 -x 开关
<code>if_smapi_bat_installed (INDEX)</code>	当使用 smapi 如果标号 INDEX 的电池装上了 显示所有

	\$if_smapi_bat_installed 到与之匹配的\$endif 之间内容
if_up (interface)	如果接口存在并且挂载上 显示所有\$if_up 到与之匹配的\$endif 之间内容
if_updatenr (updatenr)	如果这是 conky 第 updatenr 次刷新 则显示所有\$if_updatenr 到与之匹配的\$endif 之间的内容 当到达 updatenr 时计数器重置 比如 { \$if_updatenr 1 }foo\$endif{ \$if_updatenr 2 }bar\$endif{ \$if_updatenr 4 }\$endif 25%时间运行 foo 之后 25%时间运行 bar 剩余的一半时间什么都不做
if_xmms2_connected	如果 xmms2 运行了 显示所有\$if_xmms2_connected 到与之匹配的\$endif 之间内容
image <path to image> (-p x,y) (-s WxH)	在指定路径使用 IMLIB2 读取图像第二项参数 x,y 改变位置 第三项参数 w,h 表示尺寸大小 比如 \${image /home/brenden/cheeseburger.jpg -p 20,20 -s 200x200} 显示 cheeseburger.jpg 在(20,20) 尺寸为 200x200 像素
imap_messages (args)	显示 IMAP 收件箱中的邮件数量 IMAP 邮箱自己定义 形式为 host user pass [-i interval (in seconds)] [-f folder] [-p port] [-e command] [-r retries] 默认端口 143 默认文件夹 INBOX 默认时间间隔 5 分钟 默认重复次数 5 如果设置了密码 启动 conky 时候会提示输入密码
imap_unseen (args)	显示隐藏的 IMAP 收件箱中的邮箱数量 邮箱自定义 形式为 host user pass [-i interval (in seconds)] [-f folder] [-p port] [-e command] [-r retries] 默认端口 143 默认文件夹 INBOX 默认时间间隔 5 分钟 默认重复次数 5 如果设置了密码 启动 conky 时候会提示输入密码
ioscheduler disk	输出当前磁盘的硬件挂载名 比如 hda sdb
kernel	内核版本
laptop_mode	/proc/sys/vm/laptop_mode 的数值
lines textfile	显示给定文件的行数
loadavg	系统平均负荷 参数 1,2,3 1 表示过去 1 分钟 2 表示过去 5 分钟 3 表示过去 15 分钟
loadgraph ("normal" "log") (height),(width) (gradient colour 1) (gradient colour 2) (scale)	负载状态频谱图 类似于 xload 参数设置类似 cpugraph
lua function_name (function parameters)	执行一个给了参数的 lua 函数输出返回的字符串 如何加载脚本见 lua_load
lua_bar (height, width) function_name (function parameters)	执行一个给了参数的 lua 函数并以进度条形式输出预计的结果数值介于 0-100 如何加载脚本见 lua_load
lua_gauge (height, width) function_name (function parameters)	类似 lua_bar 只是将 bar 换成了 gauge
lua_graph function_name (function parameters) ("normal" "log") (height),(width) (gradient colour 1) (gradient colour 2) (scale)	执行一个给了参数的 lua 函数并以频谱图输出 预计任何输出值在默认情况都能完整表示如何加载脚本见 lua_load
lua_parse function_name (function parameters)	执行一个给了参数的 lua 函数并百分比输出按 conky 的 text 每句语法分析并输出结果数值 如何加载脚本见 lua_load
lua_read_parse function_name (conky text)	执行一个给了参数的 lua 函数并百分比输出

如果参数选择后者 则先分析每句 conky 的  
text 按 conky 的 text 每句语法分析并输出结  
果数值 如何加载脚本见 lua\_load

machine	硬件系统类型 比如 i686
mails (mailbox) (interval)	显示指定邮箱的邮件数量 邮箱支持 mbox 和 maildir 两种 可以使用像 fetchmail 的程序获得所喜欢的邮件服务器 认证 又见 new_mails
mboxscan (-n number of messages to print) (-fw from width) (-sw subject width) mbox	输出一个最近邮箱中邮件概要 邮箱参数是邮箱的文件名 比如\$ {mboxscan -n 10 "/home/brenden/some box"}
mem	使用的内存容量
membar (height),(width)	进度条显示内存的使用
memeasyfree	可用内存 包括缓冲区和缓存
memfree	剩余内存
memgauge (height),(width)	仪表显示使用的内存
memgraph ("normal" "log") (height),(width) (gradient colour 1) (gradient colour 2) (scale)	内存使用频谱图 参数设置类似 cpugraph
memmax	内存总量
memporc	内存使用百分比
mixer (device)	输出混音器状态报告 默认参数为 vol 但可以设 置"vol","bass","treble","synth","pcm","speaker","line","mic","c d","mix","pcm2","rec","igain","ogain","line1","line2","line3"," dig1","dig2","dig3","phin","phout","video","radio","monitor" 根 据 SOUND_DEVICE_NAMES 从<linux/soundcard.h> (on Linux), <soundcard.h> (on OpenBSD)或者<sys/soundcard.h>找到自己系统的参 数
mixerbar (device)	进度条显示混音器音量
mixerl (device)	混音器左声道音量
mixerlbar (device)	进度条混音器左声道音量
mixerr (device)	混音器右声道音量
mixerrbar (device)	进度条混音器右声道音量
moc_album	moc 播放歌曲的专辑
moc_artist	moc 播放歌曲的艺术家
moc_bitrate	moc 播放歌曲的比特率
moc_curtime	moc 播放歌曲的当前时间
moc_file	moc 播放歌曲的文件名
moc_rate	moc 播放歌曲的比率
moc_song	moc 播放歌曲的当前播放的歌曲名
moc_state	moc 播放歌曲的状态
moc_timeleft	moc 播放歌曲的剩余时间
moc_title	moc 播放歌曲的标题
moc_totaltime	moc 播放歌曲的总长度
monitor	conky 运行时运行的监视器数目
monitor_number	监视器的数目
mpd_album	当前播放歌曲的专辑
mpd_artist	当前播放歌曲的艺术家
mpd_bar (height),(width)	当前播放歌曲的进度条
mpd_bitrate	当前播放歌曲的比特率

mpd_elapsed	当前歌曲已播放时间
mpd_file	当前播放歌曲的文件名
mpd_length	当前播放歌曲的长度
mpd_name	当前播放歌曲的名称
mpd_percent	当前播放歌曲的播放百分比
mpd_random	mpd 播放器的随机模式(On/Off)
mpd_repeat	mpd 播放器的重复模式(On/Off)
mpd_smart (max length)	智能判断当前歌曲的输出信息, 如果 id3 可用, 则输出”歌 手-歌名”的
mpd_status	播放器状态 播放 暂停 还是停止
mpd_title (max length)	播放的歌曲标题
mpd_track	mpd 状态
mpd_vol	播放器的音量
nameserver (index)	DNS 地址 读取自/etc/resolv.conf 默认 0
new_mails (mailbox) (interval)	显示指定邮箱的邮件数量 邮箱支持 mbox 和 maildir 两种 可以使 用像 fetchmail 的程序获得所喜欢的邮件服务器认证
nodename	主机名
nvidia threshold temp ambient gpufreq memfreq imagequality	nv 显卡监视器 选项有: threshold:gpu 降速引起的温度阈值变化 temp:gpu 温度 mbient:gpu 附近气温 gpufreq:gpu 频 memfreq:gpu 内存 magequality:图像选择的 OpenGL 应用
offset (pixels)	水平向右偏移 N 像素又见 \$voffset
outlinecolor (color)	改变边缘颜色
platform (dev) type n (factor offset)	2.6 内核中 sysfs 的 platform 传感器 可以查看 /sys/bus/platform/devices/ 参数类似 hwmon
pop3_unseen (args)	显示隐藏的 POP3 收件箱中的邮箱数量 邮箱自定义 形式为 host user pass [-i interval (in seconds)] [-p port] [-e command] [-r retries] 默认 端口 110 默认文件夹 INBOX 默认时间间隔 5 分钟 默认重复 次数 5 如果设置了密码 启动 conky 时候会提示输入密码
pop3_used (args)	显示 POP3 收件箱的空间大小(2) 邮箱自定义形式为 host user pass [-i interval (in seconds)] [-p port] [-e command] [-r retries] 默认端口 110 默认文件夹 INBOX 默认时间间隔 5 分钟 默认重复次数 5 如果设置了密码 启 动 conky 时候会提示输入密码
pre_exec shell command	在 conky 显示其他任何之前执行一条命令并以文本输出
processes	正在运行的进程数(包括 sleeping 和 running)
replied_mails (maildir) (interval)	类似 draft_mails 邮箱类型为 replied
rss url delay_in_minutes action (num_par (spaces_in_front))	订阅一个 feed 并显示其最新的几个条目的标题等信息 action 可以是 feed_title,item_title,item_desc 和 item spaces_in_front 会在每个 item 前加上空格 比如: \${rss http://feed.feedsky.com/my_cnbeta 5

	item_titles 10}
	每隔五分钟获取该源的最新条目，并输出最新的十个条目的标题
running_processes	活跃的进程数
scroll length (step) text	按步滚动定长的文本 步数默认为 1 如果一个变量产生多行输出 以" "分隔 注意 不要改变颜色或其他动作影响滚动文本如果你想在文本的中间添加空格 那么在 text 的最后加上空格 比如"foobar "输出的是"foo bar"
seen_mails (maildir) (interval)	类似 draft_mails 邮箱类型为 seen
shadecolor (color)	改变阴影颜色
smapi (ARGS)	显示/sys/devices/platform/smapi 内容如果使用了 smapi
smapi_bat_bar (INDEX),(height),(width)	使用 smapi 的话 进度条显示电池剩余电量
smapi_bat_perc (INDEX)	使用 smapi 的话百分比显示电池剩余电量
smapi_bat_power INDEX	使用 smapi 的话显示电池功率
smapi_bat_temp INDEX	使用 smapi 的话显示电池温度
stippled_hr (space)	在本行剩余空间输出水平虚线 用来分隔区块 单位像素
swap	使用的交换区
swapbar (height),(width)	进度条显示交换区使用
swapmax	交换区大小
swapperc	百分比显示交换区
sysname	系统名称 比如 Linux
tab (width, (start))	start 菜单的标签 单位为像素
tail logfile lines (interval)	显示后 N 行日志文件的文本如果没设置时间间隔 默认为 2 倍 conky 时间间隔 在文本缓冲区足够的情况下最多显示 30 行
tcp_portmon port_begin port_end item (index) (ip4 only at present)	tcp 本地端口监听 只适合 ipv4 端口号 1-65535 选项有：
	count-连接总数
	rip-远程主机 IP 地址
	rhost-远程主机名
	rport-远程主机端口
	rservice-远程主机服务 见/etc/services
	lip-本地主机 IP 地址
	lhost-本地主机名
	lport-本地主机端口
	lservice-本地主机服务 见/etc/services
返回值为 0 到 n-1 大于 n-1 的值忽略 另外 一个端口不会使用多个监视器 不必担心重复的端口监视占用更多资源的问题 例如：	
	<code>\${tcp_portmon 6881 6999 count}</code> -监视 bt 端口连接数
	<code>\${tcp_portmon 22 22 rip 0}</code> -监视连接 1 的 sshd 远程主机 IP 地址
	<code>\${tcp_portmon 22 22 rip 9}</code> -监视连接 10 的 sshd 远程主机 IP 地址
	<code>\${tcp_portmon 1 1024 rhost 0}</code> -监视连接 1 的远程主机的特权端口
	<code>\${tcp_portmon 1 1024 rport 4}</code> -监视连接 5 的主机的特权端口
	<code>\${tcp_portmon 1 65535 lservice 14}</code> -监视连接 14 的本地服务名以及
打	开的所有端口
templateN (arg1) (arg2) (arg3 ...)	温度模板配置文件 N 为 0-9 允许每参数有一个空格 可以使用嵌套模板 例如：

template0 \$\1\2

template1 \1: \${fs\_used \2} / \${fs\_size \2}

template2 \1 \2

也可以不用模板 如下

使用模板	同样效果不用模板
<code>\${template0 node name}</code>	<code>\$nodename</code>
<code>\${template1 root /}</code>	<code>root: \${fs_free /} / \$ {fs_size /}</code>
<code>  \${template1   \${template2\ disk\ root}    /}</code>	<code>disk root: \${fs_free /}   / \${fs_size /}</code>

在时间间隔内运行一条命令并显示输出 和\$except类似 只不过运行于时间间隔以内如果你使用很慢的脚本在 conky 刷新期间执行 就要采用较长的时间间隔 比如你有个执行时间 5 秒的脚本 那么你的时间间隔至少要设置为 6 秒 又见\$execi

time (format)	本地时间 格式自定义 详见 strftime 的 man 文档
top type, num	按 cpu 占用自上而下排列进程 显示类型可以为 "name", "pid", "cpu", "mem" "mem_res" "mem_vsize" 和 "time" num 为标号 第 num 条 最多显示到第 10 条
top_mem type, num	和 top 类似 只不过是按内存占用排列
top_time type, num	和 top 类似 只不过是按 cpu 时序排列
totaldown net	下载总量 从 conky 运行开始计数 32 位系统超出的无法显示
totalup net	上传总量 和 totaldown 类似
trashed_mails (maildir) (interval)	类似 draft_mails 邮箱类型为 trashed
tztime (timezone) (format)	本地时间(指定拾取)见/usr/share/zoneinfo 比如 US/Pacific 和 Europe/Zurich
unflagged_mails (maildir) (interval)	类似 draft_mails 邮箱类型为 unflagged
unforwarded_mails (maildir) (interval)	类似 draft_mails 邮箱类型为 unforwarded
unreplied_mails (maildir) (interval)	类似 draft_mails 邮箱类型为 unreplied
unseen_mails (maildir) (interval)	类似 draft_mails 邮箱类型为 unseen
updates Number of updates	conky 的刷新次数 一般调试时用的
upspeed net	上传速度 单位 KiB
upspeedf net	带小数位的上传速度 单位 KiB
upspeedgraph (netdev) ("normal" "log") (height),(width) (gradient colour 1) (gradient colour 2) (scale)	上传速度频谱图 参数和 cpugraph 相似
uptime	电脑运行时间
uptime_short	uptime 的缩简版
user_names	列出登录的用户
user_number	登录的用户数量
user_terms	列出使用的终端
user_times	列出用户登录得时间
utime (format)	UTC 时间
voffset (pixels)	垂直向上偏移 N 像素负值向下偏移 又见\$offset
voltage_mv (n)	返回 cpu 电压值 单位 mV 参数为 cpu 标号 默认 1
voltage_v (n)	返回 cpu 电压值 单位 V 参数为 cpu 标号 默认 1

wireless_ap net	无线 AP 的 MAC 地址 仅适用于 Linux
wireless_bitrate net	无线连接速率 如 11Mb/s 仅适用于 Linux
wireless_essid net	无线 AP 名称 ESSID 仅适用于 Linux
wireless_link_bar (height), (width) net	进度条无线连接状态仅适用于 Linux
wireless_link_qual net	无线连接信号强度 仅适用于 Linux
wireless_link_qual_max net	无线连接信号强度最大值 仅适用于 Linux
wireless_link_qual_perc net	百分比显示无线连接信号强度 仅适用于 Linux
wireless_mode net	无线模式(Managed/Ad-Hoc/Master) 仅适用于 Linux
words textfile	显示给定文件中的单词数