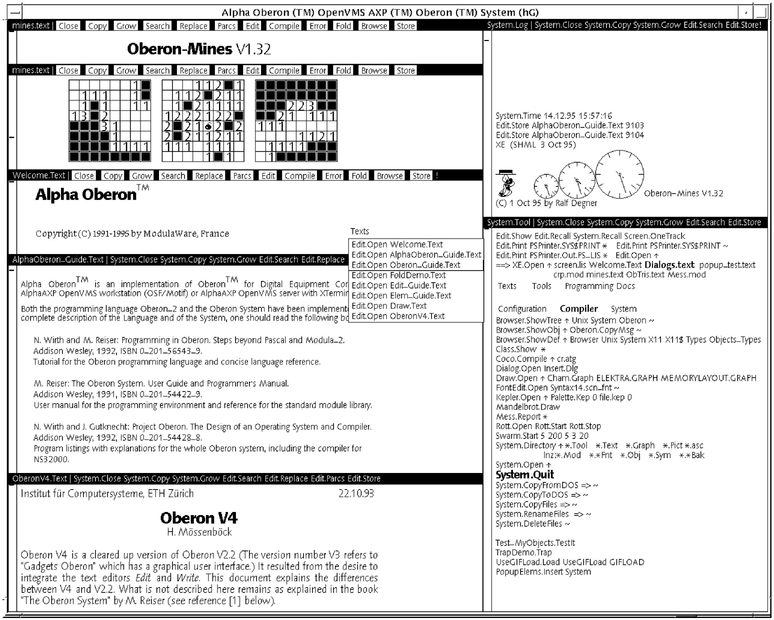
abduco & dvtm

Session and Tiling Window Management for the Console

Marc André Tanner

CoSin '18

Oberon



Alpha Oberon (TM) OpenVMS AXP (TM) Oberon (TM) System (hG)

mines.text | Close Copy Grow Search Replace Parcs Edit Compile Error Fold Browse Store

Oberon-Mines V1.32

mines.text | Close Copy Grow Search Replace Parcs Edit Compile Error Fold Browse Store

Welcome.Text | Close Copy Grow Search Replace Parcs Edit Compile Error Fold Browse Store !

Alpha Oberon™

Copyright (C) 1991-1995 by ModulaWare, France

Texts
Edit.Open Welcome.Text
Edit.Open AlphaOberon_Guide.Text
Edit.Open Oberon_Guide.Text
Edit.Open FoldDemo.Text
Edit.Open Edit_Guide.Text
Edit.Open Elem_Guide.Text
Edit.Open Draw.Text
Edit.Open OberonV4.Text

AlphaOberon_Guide.Text | System.Close System.Copy System.Grow Edit.Search Edit.Replace

Alpha Oberon™ is an implementation of Oberon™ for Digital Equipment Cor
AlphaAXP OpenVMS workstation (OSF/Motif) or AlphaAXP OpenVMS server with XTermi

Both the programming language Oberon_2 and the Oberon System have been implement
complete description of the Language and of the System, one should read the following bo

N. Wirth and M. Reiser: Programming in Oberon. Steps beyond Pascal and Modula_2.
Addison Wesley, 1992, ISBN 0_201_56543_9.
Tutorial for the Oberon programming language and concise language reference.

M. Reiser: The Oberon System. User Guide and Programmer's Manual.
Addison Wesley, 1991, ISBN 0_201_54422_9.
User manual for the programming environment and reference for the standard module library.

N. Wirth and J. Gutknecht: Project Oberon. The Design of an Operating System and Compiler.
Addison Wesley, 1992, ISBN 0_201_54428_8.
Program listings with explanations for the whole Oberon system, including the compiler for NS32000.

OberonV4.Text | System.Close System.Copy System.Grow Edit.Search Edit.Replace Edit.Parcs Edit.Store

Institut für Computersysteme, ETH Zürich 22.10.93

Oberon V4
H. Mössenböck

Oberon V4 is a cleared up version of Oberon V2.2 (The version number V3 refers to "Gadgets Oberon" which has a graphical user interface.) It resulted from the desire to integrate the text editors *Edit* and *Write*. This document explains the differences between V4 and V2.2. What is not described here remains as explained in the book "The Oberon System" by M. Reiser (see reference [1] below).

System.Log | System.Close System.Copy System.Grow Edit.Search Edit.Store!

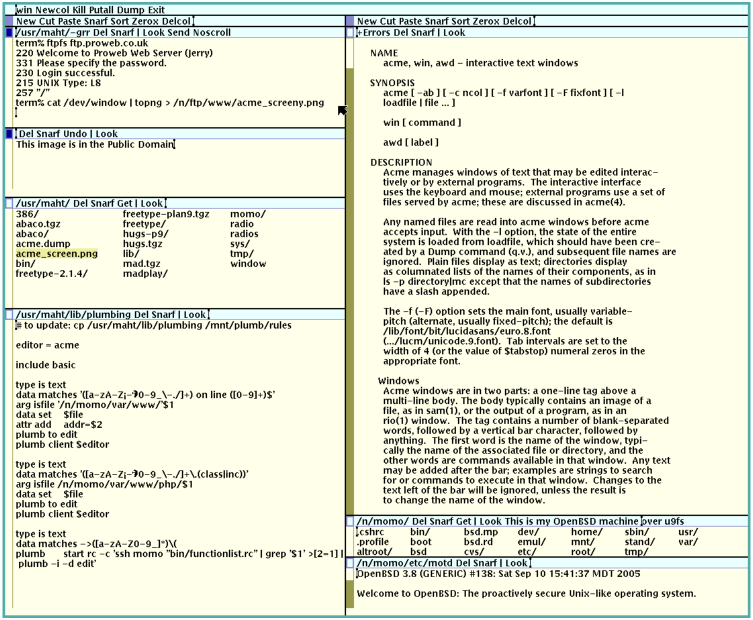
```
System.Time 14.12.95 15:57:16
Edit.Store AlphaOberon_Guide.Text 9103
Edit.Store AlphaOberon_Guide.Text 9104
XE  (SHML  3 Oct 95)
```

Oberon-Mines V1.32
(C) 1 Oct 95 by Ralf Degner

System.Tool | System.Close System.Copy System.Grow Edit.Search Edit.Store

```
Edit.Show Edit.Recall System.Recall Screen.OneTrack
Edit.Print PSPrinter.SYS$PRINT *   Edit.Print PSPrinter.SYS$PRINT ~
Edit.Print PSPrinter.Out.PS_LIS *   Edit.Open ↑
==> XE.Open ↑ screen.lis Welcome.Text Dialogs.text popup_test.text
        crp.mod mines.text ObTris.text Mess.mod
 Texts    Tools    Programming Docs

 Configuration    Compiler    System
Browser.ShowTree ↑ Unix System Oberon ~
Browser.ShowObj ↑ Oberon.CopyMsg ~
Browser.ShowDef ↑ Browser Unix System X11 X11$ Types Objects_Types
Class.Show *
Coco.Compile ↑ cr.atg
Dialog.Open Insert.Dlg
Draw.Open ↑ Cham.Graph ELEKTRA.GRAPH MEMORYLAYOUT.GRAPH
FontEdit.Open Syntax14.scn_fnt ~
Kepler.Open ↑ Palette.Kep 0 file.kep 0
Mandelbrot.Draw
Mess.Report *
Rott.Open Rott.Start Rott.Stop
Swarm.Start 5 200 5 3 20
System.Directory ↑ *.Tool   *.Text   *.Graph   *.Pict *.asc
        lnz:*.Mod   *.*Fnt   *.Obj   *.Sym   *.*Bak
System.Open ↑
System.Quit
System.CopyFromDOS => ~
System.CopyToDOS => ~
System.CopyFiles => ~
System.RenameFiles  => ~
System.DeleteFiles ~

Test_MyObjects.TestIt
TrapDemo.Trap
UseGIFLoad.Load UseGIFLoad GIFLOAD
PopupElems.Insert System
```

Acme



```
win Newcol Kill Putall Dump Exit
```

```
New Cut Paste Snarf Sort Zerox Delcol
```

```
/usr/maht/-grr Del Snarf | Look Send Noscroll
term% ftpfs ftp.proweb.co.uk
220 Welcome to Proweb Web Server (Jerry)
331 Please specify the password.
230 Login successful.
215 UNIX Type: L8
257 "/"
term% cat /dev/window | topng > /n/ftp/www/acme_screeny.png
```

```
Del Snarf Undo | Look
This image is in the Public Domain
```

```
/usr/maht/ Del Snarf Get | Look
386/                freetype-plan9.tgz    momo/
abaco.tgz           freetype/             radio
abaco/              hugs-p9/              radios
acme.dump           hugs.tgz              sys/
acme_screen.png     lib/                  tmp/
bin/                mad.tgz               window
freetype-2.1.4/     madplay/
```

```
/usr/maht/lib/plumbing Del Snarf | Look
# to update: cp /usr/maht/lib/plumbing /mnt/plumb/rules

editor = acme

include basic

type is text
data matches '([a-zA-Z¡-￾0-9_\-./]+) on line ([0-9]+)$'
arg isfile '/n/momo/var/www/'$1
data set    $file
attr add    addr=$2
plumb to edit
plumb client $editor

type is text
data matches '([a-zA-Z¡-￾0-9_\-./]+\.(class|inc))'
arg isfile /n/momo/var/www/php/$1
data set    $file
plumb to edit
plumb client $editor

type is text
data matches ->([a-zA-Z0-9_]*)\(
plumb       start rc -c 'ssh momo "bin/functionlist.rc" | grep '$1' >[2=1] |
 plumb -i -d edit'
```

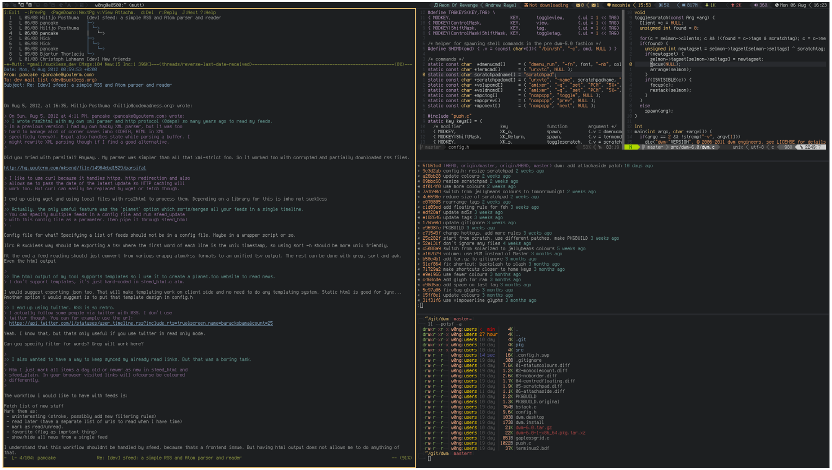
```
New Cut Paste Snarf Sort Zerox Delcol
```

```
+Errors Del Snarf | Look

  NAME
       acme, win, awd – interactive text windows

  SYNOPSIS
       acme [ -ab ] [ -c ncol ] [ -f varfont ] [ -F fixfont ] [ -l
       loadfile | file ... ]

       win [ command ]

       awd [ label ]

  DESCRIPTION
       Acme manages windows of text that may be edited interac-
       tively or by external programs.  The interactive interface
       uses the keyboard and mouse; external programs use a set of
       files served by acme; these are discussed in acme(4).

       Any named files are read into acme windows before acme
       accepts input.  With the -l option, the state of the entire
       system is loaded from loadfile, which should have been cre-
       ated by a Dump command (q.v.), and subsequent file names are
       ignored.  Plain files display as text; directories display
       as columnated lists of the names of their components, as in
       ls -p directory|mc except that the names of subdirectories
       have a slash appended.

       The -f (-F) option sets the main font, usually variable-
       pitch (alternate, usually fixed-pitch); the default is
       /lib/font/bit/lucidasans/euro.8.font
       (.../lucm/unicode.9.font).  Tab intervals are set to the
       width of 4 (or the value of $tabstop) numeral zeros in the
       appropriate font.

     Windows
       Acme windows are in two parts: a one-line tag above a
       multi-line body. The body typically contains an image of a
       file, as in sam(1), or the output of a program, as in an
       rio(1) window.  The tag contains a number of blank-separated
       words, followed by a vertical bar character, followed by
       anything.  The first word is the name of the window, typi-
       cally the name of the associated file or directory, and the
       other words are commands available in that window.  Any text
       may be added after the bar; examples are strings to search
       for or commands to execute in that window.  Changes to the
       text left of the bar will be ignored, unless the result is
       to change the name of the window.
```

```
/n/momo/ Del Snarf Get | Look This is my OpenBSD machine over u9fs
.cshrc      bin/       bsd.mp     dev/       home/      sbin/      usr/
.profile    boot       bsd.rd     emul/      mnt/       stand/     var/
altroot/    bsd        cvs/       etc/       root/      tmp/
```

```
/n/momo/etc/motd Del Snarf | Look
OpenBSD 3.8 (GENERIC) #138: Sat Sep 10 15:41:37 MDT 2005

Welcome to OpenBSD: The proactively secure Unix-like operating system.
```

X11: wmi, wmii, dwm



```
w0ng@e8500:~ (mutt)

i:Exit  -:PrevPg  <PageDown>:NextPg v:View Attachm.  d:Del  r:Reply  J:Next ?:Help
  1   L 05/08 Hiltjo Posthuma   [dev] sfeed: a simple RSS and Atom parser and reader
  2   L 06/08 pancake           ├─>
  3   L 06/08 Hiltjo Posthuma   │ └─>
  4   L 06/08 pancake           │   └─>
  5   L 06/08 Nick              ├─>
  6   L 06/08 Nick              │ └─>
  7   L 06/08 pancake           │   └─>
  8   L 05/08 Bjartur Thorlaciu └─>
  9   L 01/08 Christoph Lohmann [dev] New friends
-*-Mutt: =gmail/suckless_dev [Msgs:104 New:15 Inc:1 396K]---(threads/reverse-last-date-received)----------------(8%)---
Date: Mon, 6 Aug 2012 00:59:53 +0200
From: pancake <pancake@youterm.com>
To: dev mail list <dev@suckless.org>
Subject: Re: [dev] sfeed: a simple RSS and Atom parser and reader



On Aug 5, 2012, at 16:35, Hiltjo Posthuma <hiltjo@codemadness.org> wrote:

> On Sun, Aug 5, 2012 at 4:11 PM, pancake <pancake@youterm.com> wrote:
>> I wrote rss2html with my own xml parser and http protocol (0deps) so many years ago to read my feeds.
> In a previous version I had my own hacky XML parser, but it was too
> hard to manage alot of corner cases imho (CDATA, HTML in XML
> specificly (eeew)). Expat also handles state while parsing a buffer. I
> might rewrite XML parsing though if I find a good alternative.
>

Did you tried with parsifal? Anyway.. My parser was simpler than all that xml-strict foo. So it worked too with corrupted and partially downloaded rss files.

http://hq.youterm.com/mksend/file/14984ebd1529/parsifal

> I like to use curl because it handles https, http redirection and also
> allows me to pass the date of the latest update so HTTP caching will
> work too. But curl can easily be replaced by wget or fetch though.

I end up using wget and using local files with rss2html to process them. Depending on a library for this is imho not suckless
>
>> Actually, the only useful feature was the 'planet' option which sorts/merges all your feeds in a single timeline.
> You can specify multiple feeds in a config file and run sfeed_update
> with this config file as a parameter. Then pipe it through sfeed_html
> .

Config file for what? Specifying a list of feeds should not be in a config file. Maybe in a wrapper script or so.

Iirc A suckless way should be exporting a tsv where the first word of each line is the unix timestamp, so using sort -n should be more unix friendly.

At the end a feed reading should just convert from various crappy atom/rss formats to an unified tsv output. The rest can be done with grep, sort and awk.
Even the html output

>
>> The html output of my tool supports templates so i use it to create a planet.foo website to read news.
> I don't support templates, it's just hard-coded in sfeed_html.c atm.

I would suggest exporting json too. That will make templating work on client side and no need to do any templating system. Static html is good for lynx...
Another option i would suggest is to put that template design in config.h
>
>> I end up using twitter. RSS is so retro.
> I actually follow some people via twitter with RSS. I don't use
> twitter though. You can for example use the url:
> https://api.twitter.com/1/statuses/user_timeline.rss?include_rts=true&screen_name=barackobama&count=25

Yeah. I know that, but thats only useful if you use twitter in read only mode.

Can you specify filter for words? Grep will work here?

>
>> I also wanted to have a way to keep synced my already read links. But that was a boring task.
>
> Atm I just mark all items a day old or newer as new in sfeed_html and
> sfeed_plain. In your browser visited links will ofcourse be coloured
> differently.
>

The workflow i would like to have with feeds is:

Fetch list of new stuff
Mark them as:
 - uninteresting (stroke, possibly add new filtering rules)
 - read later (have a separate list of urls to read when i have time)
 - mark as read/unread.
 - favorite (flag as imprtant thing)
 - show/hide all news from a single feed

I understand that this workflow shouldnt be handled by sfeed, because thats a frontend issue. But having html output does not allows me to do anything of
that.
-  L- 4/104: pancake            Re: [dev] sfeed: a simple RSS and Atom parser and reader                    -- (91%)
```

```
♫ Aeon Of Revenge < Andrew Rayel   Not downloading   0 < 1   mooshie < 15:53   5%  < 817M   1K   2K   36%   Mon 06 Aug < 16:23
```

```
#define TAGKEYS(KEY,TAG) \
{ MODKEY,                           KEY,      toggleview,     {.ui = 1 << TAG} \
{ MODKEY|ControlMask,               KEY,      view,           {.ui = 1 << TAG} \
{ MODKEY|ShiftMask,                 KEY,      tag,            {.ui = 1 << TAG} \
{ MODKEY|ControlMask|ShiftMask, KEY,      toggletag,      {.ui = 1 << TAG} \

/* helper for spawning shell commands in the pre dwm-5.0 fashion */
#define SHCMD(cmd) { .v = (const char*[]){ "/bin/sh", "-c", cmd, NULL } }

/* commands */
static const char  *dmenucmd[]     = { "dmenu_run", "-fn", font, "-nb", col
static const char *termcmd[]       = { "urxvtc", NULL };
static const char scratchpadname[] = "scratchpad";
static const char *scratchpadcmd[] = { "urxvtc", "-name", scratchpadname, "
static const char *volupcmd[]      = { "amixer", "-q", "set", "PCM", "5%+",
static const char *voldncmd[]      = { "amixer", "-q", "set", "PCM", "5%-",
static const char *mpctog[]        = { "ncmpcpp", "toggle", NULL };
static const char *mpcprev[]       = { "ncmpcpp", "prev", NULL };
static const char *mpcnext[]       = { "ncmpcpp", "next", NULL };

#include "push.c"
static Key keys[] = {
  /* modifier                     key              function        argument */
  { MODKEY,                       XK_o,            spawn,          {.v = dmenucm
  { MODKEY|ShiftMask,             XK_Return,       spawn,          {.v = termcmd
  { MODKEY,                       XK_s,            togglescratch,  {.v = scratch
 master  config.h                                                    53% <  83:19
```

```
void
togglescratch(const Arg *arg) {
  Client *c = NULL;
  unsigned int found = 0;

  for(c = selmon->clients; c && !(found = c->tags & scratchtag); c = c->ne
  if(found) {
    unsigned int newtagset = selmon->tagset[selmon->seltags] ^ scratchtag;
    if(newtagset) {
      selmon->tagset[selmon->seltags] = newtagset;
      focus(NULL);
      arrange(selmon);
    }
    if(ISVISIBLE(c)) {
      focus(c);
      restack(selmon);
    }
  }
  else
    spawn(arg);
}

int
main(int argc, char *argv[]) {
  if(argc == 2 && !strcmp("-v", argv[1]))
    die("dwm-"VERSION", © 2006-2011 dwm engineers, see LICENSE for details
 N   master > src/dwm-6.0/dwm.c                   unix < utf-8 < c   98%   2249:7
```

```
* 5fb51c4 (HEAD, origin/master, origin/HEAD, master) dwm: add attachaside patch 10 days ago
* 9c3d2ab config.h: resize scratchpad 2 weeks ago
* a26bb28 update colours 2 weeks ago
* 89bbc68 resize scratchpad 2 weeks ago
* df014f0 use more colours 2 weeks ago
* 7afb90d switch from jellybeans colours to tomorrownight 2 weeks ago
* 4c6598e reduce size of scratchpad 2 weeks ago
* e070805 rearrange tags 2 weeks ago
* c1d09ed add floating rule for feh 3 weeks ago
* edf28af update md5s 3 weeks ago
* e182646 update tags 3 weeks ago
* 175be0d update gitignore 3 weeks ago
* e9698fe PKGBUILD 3 weeks ago
* c71549f change hotkeys, add more rules 3 weeks ago
* 25c282f start from scratch, use different patches, make PKGBUILD 3 weeks ago
* 52e131f don't ignore any files 4 weeks ago
* c5088a9 switch from solarized to jellybeans colours 5 weeks ago
* a107b29 volume: use PCM instead of Master 3 months ago
* b50c4b1 add tar.gz to gitignore 3 months ago
* 91ef864 fix shortcut: backslash to slash 3 months ago
* 71729a2 make shortcuts closer to home keys 3 months ago
* e9e1966 use fewer colours 3 months ago
* c468cde add glyph for ram 3 months ago
* c98d5ac add space on last tag 3 months ago
* 5c97a06 fix tag glyphs 3 months ago
* 15ff0e1 update colours 3 months ago
* 31f31f6 use vimpowerline glyphs 3 months ago
:
```

```
 ~/git/dwm  master=
 ll --potsf -a
 drwxr-xr-x w0ng:users <  min     4K .
 drwxr-xr-x w0ng:users 27 hour    4K ..
 drwxr-xr-x w0ng:users 10 day     4K .git
 drwxr-xr-x w0ng:users 10 day     4K pkg
 drwxr-xr-x w0ng:users 10 day     4K src
 -rw-r--r-- w0ng:users 14 sec    16K .config.h.swp
 -rw-r--r-- w0ng:users 19 day    30B .gitignore
 -rw-r--r-- w0ng:users 14 day   7.6K 01-statuscolours.diff
 -rw-r--r-- w0ng:users 19 day   1.2K 02-monoclecount.diff
 -rw-r--r-- w0ng:users 19 day   2.6K 03-noborder.diff
 -rw-r--r-- w0ng:users 19 day   1.7K 04-centredfloating.diff
 -rw-r--r-- w0ng:users 19 day   1.9K 05-scratchpad.diff
 -rw-r--r-- w0ng:users 11 day   1.1K 06-attachaside.diff
 -rw-r--r-- w0ng:users 10 day   2.2K PKGBUILD
 -rw-r--r-- w0ng:users 10 day   1.3K PKGBUILD.original
 -rw-r--r-- w0ng:users 19 day   764B bstack.c
 -rw-r--r-- w0ng:users 10 day   9.6K config.h
 -rw-r--r-- w0ng:users 19 day   103B dwm.desktop
 -rw-r--r-- w0ng:users 19 day   173B dwm.install
 -rw-r--r-- w0ng:users 10 day    21K dwm-6.0.tar.gz
 -rw-r--r-- w0ng:users 10 day    22K dwm-6.0-1-x86_64.pkg.tar.xz
 -rw-r--r-- w0ng:users 19 day   851B gaplessgrid.c
 -rw-r--r-- w0ng:users 19 day  1022B push.c
 -rw-r--r-- w0ng:users 19 day    37K terminus2.bdf
 ~/git/dwm  master=
```

TL;DR: What is this all about?

Provide a similiar working environment suitable for:

- Framebuffer console
- Remote e.g. SSH/mosh sessions

TL;DR: What is this all about?

Provide a similiar working environment suitable for:

- Framebuffer console
- Remote e.g. SSH/mosh sessions

abduco

- session persistence

TL;DR: What is this all about?

Provide a similiar working environment suitable for:

- Framebuffer console
- Remote e.g. SSH/mosh sessions

abduco

- session persistence

dvtm

- tiling window management for the console
- terminal multiplexer

Tiling window management

- Optimally use available screen space
- No overlapping windows
- Automatic window placement
- Minimal window decorations

dvtm – *dynamic* virtual terminal manager

```
From: Marc Andre Tanner <mat@brain-dump.org>
Date: Sat, 8 Dec 2007 13:29:30 +0100
To: dwm@suckless.org
Subject: [ANNOUNCE] dvtm - dynamic virtual terminal
 manager - aka dwm for the console

Hi,

For some time I have been thinking about applying
the concept of tiling window management to the
console. As a result I have written dvtm, you can
check it out here:

 http://www.brain-dump.org/projects/dvtm/

...
```

Concepts shared with `dwm`

- window management should be automatic and *dynamic*
- master and stacking area
- tagging concept
- similar key bindings, `MOD` defaults to `Ctrl-g`
- 1-line statusbar (via a named pipe)
- configuration through `config.def.h`

Design Philosophy

Heavily influenced by `suckless.org`.

Focus on simplicity, clarity and frugality, minimal but useable, do one thing and do it well.

- *dynamic* window management for the console
- no internal copy mode (use `$EDITOR` instead)
- no session support (see `abduco`)
- easily scriptable

dvtm – *dynamic* virtual terminal manager

Single *modifier* key, prefix for all commands

Denoted by `$MOD`, defaults to ⟨`C-g`⟩

dvtm – *dynamic* virtual terminal manager

Single *modifier* key, prefix for all commands

Denoted by `$MOD`, defaults to ⟨`C-g`⟩

Can be changed at runtime:

- `dvtm -m ^a` (⟨`C-a`⟩ as in screen)
- `dvtm -m ^b` (⟨`C-b`⟩ as in tmux)

Use `$MOD`-`$MOD` to send the `$MOD` key.

dvtm – window lifecycle

- `dvtm` process1 process2 …
- `$MOD-c` create new window
- `$MOD-C` create new window with same working directory[1]
- `$MOD-x-x` close window

Closing the last window, terminates dvtm.

[1] Depends on `/proc/$PID/cwd`

dvtm – focus windows

- `$MOD-j` focus next
- `$MOD-k` focus previous
- `$MOD-J` focus next non-minimized
- `$MOD-K` focus previous non-minimized
- `$MOD-`[0..9] focus *n*-th window
- `$MOD-`⟨`Tab`⟩ focus previously selected window

dvtm – master and stacking area

Available space is split into two areas:

- master: the primary window(s)
- stacking: the other windows

dvtm – change master area

- `$MOD-⟨Enter⟩` swap current window to/from master area

dvtm – change master area

- `$MOD-⟨Enter⟩` swap current window to/from master area
- `$MOD-l` increase master area width
- `$MOD-h` decrease master area width

dvtm – change master area

- `$MOD-⟨Enter⟩` swap current window to/from master area
- `$MOD-l` increase master area width
- `$MOD-h` decrease master area width
- `$MOD-i` increase number of windows in master area
- `$MOD-d` decrease number of windows in master area

dvtm – minimize/maximize windows

- `$MOD-.` toggle minimization of current window
- `$MOD-m` maximize current window

dvtm – layouts

A way to place/display windows.

`$MOD-`⟨`Space`⟩ cycles through layouts.

dvtm – layouts

A way to place/display windows.

`$MOD-⟨Space⟩` cycles through layouts.

- `$MOD-f` Vertical stack (default)
- `$MOD-b` Bottom stack
- `$MOD-g` Grid
- `$MOD-m` Monocle/fullscreen

dvtm – layouts

A way to place/display windows.

`$MOD-⟨Space⟩` cycles through layouts.

- `$MOD-f` Vertical stack (default)
- `$MOD-b` Bottom stack
- `$MOD-g` Grid
- `$MOD-m` Monocle/fullscreen

Also included in source tarball, but disabled by default:

- Top stack
- Vertical stack
- Fibonacci: spiral & dwindle

dvtm – tagging concept

Controls which windows are displayed.

A super set of the workspace functionality.

dvtm – tagging concept

Controls which windows are displayed.

A super set of the workspace functionality.

A static set of $tags = \{tag1, tag2, \ldots, tagN\}$

Every window is tagged with at least one tag.

dvtm – tagging concept

Controls which windows are displayed.

A super set of the workspace functionality.

A static set of $tags = \{tag1, tag2, \ldots, tagN\}$

Every window is tagged with at least one tag.

A *view* is a subset of *tags* i.e. $(views \subseteq tags)$

A *view* displays all windows having at least one of the tags.

dvtm – tagging modifiing the view

View tag: display all windows with *tagN*, "change workspace"

- `$MOD-v-`*N*

dvtm – tagging modifiing the view

View tag: display all windows with *tagN*, "change workspace"

- `$MOD-v-`*N*

Toggle tag of view: add/remove all windows with *tagN*

- `$MOD-V-`*N*

dvtm – tagging windows

Tag window: apply *tagN* to focused window, "move window to workspace"

- `$MOD-t-`*N*

dvtm – tagging windows

Tag window: apply *tagN* to focused window, "move window to workspace"

- `$MOD-t-`*N*

Toggle tag of window: add/remove *tagN* from focused window

- `$MOD-T-`*N*

dvtm – miscellaneous tagging

`$MOD-v-⟨Tab⟩` switch to previously selected tags

`$MOD-0` view all tags / windows

dvtm – status bar

Hidden by default.

Displays a single line of text, read from a FIFO:

dvtm – status bar

Hidden by default.

Displays a single line of text, read from a FIFO:

- `mkfifo -m 600 dvtm.status`
- `dvtm -s dvtm.status`
- `echo "your fancy status" > dvtm.status`

See `dvtm-status(1)` for an extended example

dvtm – status bar

Hidden by default.

Displays a single line of text, read from a FIFO:

- `mkfifo -m 600 dvtm.status`
- `dvtm -s dvtm.status`
- `echo "your fancy status" > dvtm.status`

See `dvtm-status(1)` for an extended example

`$MOD-s` toggles status bar

`$MOD-S` cylces position (top, bottom)

dvtm – scrollback buffer

Enhances terminals like `st(1)` with a scroll back buffer.

- Set history size: `dvtm -h lines`
- ⟨`S-PageUp`⟩ or `$MOD+`⟨`PageUp`⟩ scroll up
- ⟨`S-PageDown`⟩ or `$MOD+`⟨`PageDown`⟩ scroll down

dvtm – keyboard multiplexing

Keypresses are forwarded to all visible windows.

Useful to interactively manage multiple servers.

- `$MOD-a` toggles multiplexing mode

dvtm – copymode

Copy and paste text across windows.

- uses your `$EDITOR` as interactive filter
    - pipes scroll back buffer history to editor
    - keeps whatever the editor writes to stdout in a register
    - `dvtm-editor(1)` makes it work for ordinary `$EDITOR`s

dvtm – copymode

Copy and paste text across windows.

- uses your `$EDITOR` as interactive filter
  - pipes scroll back buffer history to editor
  - keeps whatever the editor writes to stdout in a register
  - `dvtm-editor(1)` makes it work for ordinary `$EDITOR`s

- `$MOD-e` enter copy mode
- `$MOD-p` paste previously copied text

dvtm – window title

Xterm terminal escape sequence extension:

```
$ printf "\033]0;%s\007" "Your title here!"
```

See also `dvtm-title(1)`

dvtm – urgent flag

Titlebar (or tagbar) indication that "something" occured in the window.

Triggered by ASCII bell character `\a`.

dvtm – mouse support

Click to focus window.

Double click to focus and toggle maximization.

Middle click to zoom.

Rright click to minimize.

dvtm – scripting capabilities

Control dvtm from other processes.

Reads commands from a named pipe.

dvtm – scripting capabilities

Control dvtm from other processes.

Reads commands from a named pipe.

- `dvtm -c dvtm-command.fifo`
- `echo "create vis" > dvtm-command.fifo`
- `$DVTM_CMD_FIFO` exposed to child processes

dvtm – scripting capabilities

Control dvtm from other processes.

Reads commands from a named pipe.

- `dvtm -c dvtm-command.fifo`
- `echo "create vis" > dvtm-command.fifo`
- `$DVTM_CMD_FIFO` exposed to child processes

Only unidirectional communication.

Still limited and experimental.

abduco: session handling

- Provides session persistence
  - terminate stuck SSH sessions ⟨`Enter`⟩ `~` `.`
  - ssh user@host -t abduco -A session

abduco: session handling

- Provides session persistence
  - terminate stuck SSH sessions ⟨`Enter`⟩ `~ .`
  - ssh user@host -t abduco -A session

- Simple client/server architecture
- Communication over Unix domain socket

abduco: session handling

- Provides session persistence
  - terminate stuck SSH sessions ⟨`Enter`⟩ `~ .`
  - ssh user@host -t abduco -A session

- Simple client/server architecture
- Communication over Unix domain socket

- Operates on the raw I/O stream
- Does not attempt to interpret or preserve terminal state

abduco: basic usage

Create session (and attach)

- `abduco -c demo`

abduco: basic usage

Create session (and attach)

- `abduco -c demo`

Detach session

- ⟨`Ctrl-\`⟩

abduco: basic usage

Create session (and attach)

- `abduco -c demo`

Detach session

- ⟨`Ctrl-\`⟩

Reattach session

- `abduco -a demo`

abduco: session list

```
$ abduco
Active sessions (on host thinkpad)
* Sat 2018-06-16 20:40:36  27492  connected
  Sat 2018-06-16 20:39:49  27414  inactive
+ Sat 2018-06-16 20:40:13  27487  dead
```

abduco: session list

```
$ abduco
Active sessions (on host thinkpad)
* Sat 2018-06-16 20:40:36   27492   connected
  Sat 2018-06-16 20:39:49   27414   inactive
+ Sat 2018-06-16 20:40:13   27487   dead
```

Column meaning:

1. Status, $*$ active / client connected, + terminated
2. Last activity (`mtime` of socket)
3. Server PID
4. Session name

abduco: session exit status

No output buffering, but exit status is recorded.

```
$ abduco -n demo false && abduco -a demo
abduco: demo: session terminated
                              with exit status 1
```

abduco: shared sessions

Multiple simultaneously connected clients.

- Most recently non-readonly client dictates pty size
- Read only sessions (input is discarded)
- For security purposes, use `socat(1)`

```
$ socat -u unix-connect:/tmp/abduco/private/session
           unix-listen:/tmp/abduco/public/read-only &
```

abduco: resize handling

Most recently non-readonly client dictates `pty(7)` size.

Delivers `SIGWINCH` to underlying process.

abduco: socket recreation

In case session socket disappears:

- `pgrep -P 1 abduco`
- `lsof -p $PID | grep unix`
- `kill -USR1 $PID`
- `cp /proc/$PID/exe abduco`
- `./abduco`

abduco: environment variables

Command to run, if omitted:

- `$ABDUCO_CMD` defaults to `dvtm`

Current session information:

- `$ABDUCO_SESSION`
- `$ABDUCO_SOCKET`

Limitations & a plan to fix them

Terminal state not preserved across sessions

Possible fix:

1. session attached
2. abduco sends signal to supervised application (i.e. dvtm)
3. dvtm restores terminal state

Future Plans[2]

- Find more time for maintenance
- Preserve terminal state across sessions
- Improve terminal emulation
  - 24 bit color support
  - `dvtm` $\approx$ `dwm` + `st` ?
  - `dvtm` $\approx$ `libvterm` + `libtickit` ?

[2] In no particular order, no timeline given.

Future Plans

- Improve scripting capabilities, allow bidirectional communication via a unix domain socket

```
$ echo cmd | socat - UNIX-CONNECT:/tmp/socket | doit
```

- Provide Lua API?
- Resolve abduco license controversy

Conclusion

Does not conflate session and window managment.

Allthough raw edges, conceptually sound.

Non-bloated solution which works (at least for my usecase).

Questions?

https://github.com/martanne/abduco
https://github.com/martanne/dvtm

git://repo.or.cz/abduco.git
git://repo.or.cz/dvtm.git

mat@brain-dump.org

#vis-editor on freenode

Happy Hacking!