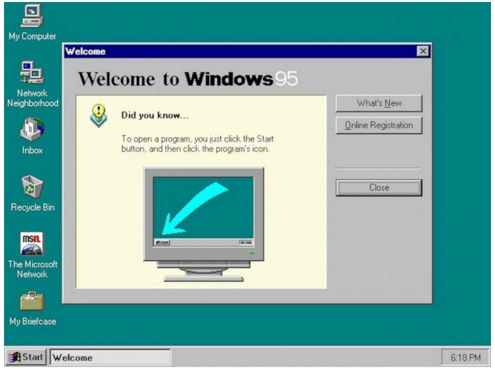
Desktop Environments, Window Managers and Ricing (Oh My!)

- Featuring AwesomeWM and Hyprland

What is a Desktop Environment?

- A desktop environment (DE) is an extensive GUI that allows to intuitively interact with their operating system and the various applications and tools currently installed.
- They are composed of a multitude of individual components such as a window manager, file manager, taskbars, panels and widgets that provide streamlined access to various features of your system.
- The design of a desktop environment is often derived from the idea of it being an actual desktop, with windows resting on top of eachother like papers and having all of your utilities at an arm's reach.

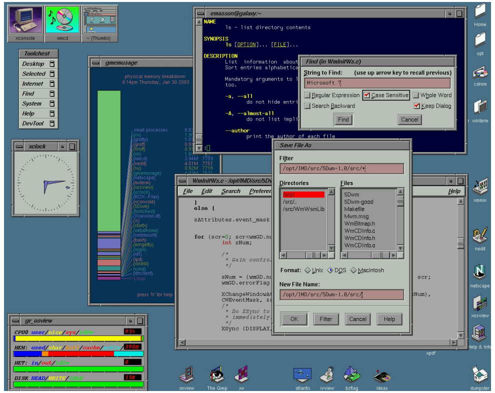


Welcome to Windows 95

Did you know...

To open a program, you just click the Start button, and then click the program's icon.

(4Dwm environment from the Silicon Graphics workstations running IRIX.)



```
NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       List  information  about
       Sort entries alphabetica

       Mandatory arguments to l
       too.

       -a, --all
              do not hide entri

       -A, --almost-all
              do not list impli

       --author
              print the author of each file
:
```

Find (in WmInitWs.c)

String to Find: (use up arrow key to recall previous)

Microsoft ?

□ Regular Expression ☑ Case Sensitive □ Whole Word

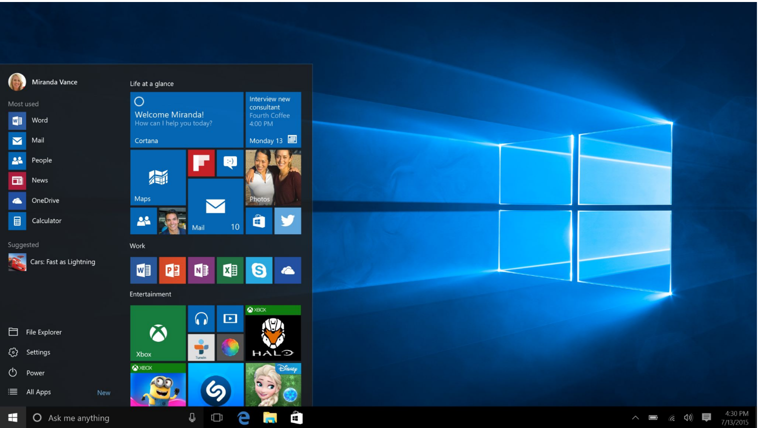
□ Search Backward ☑ Keep Dialog

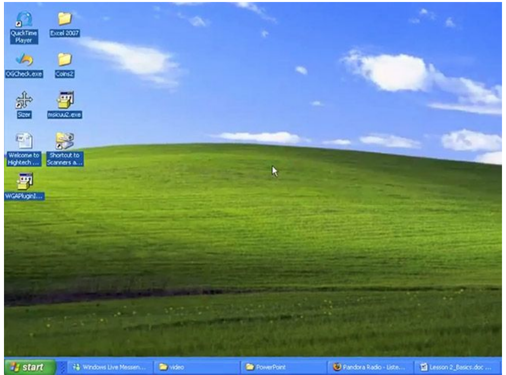
Save File As

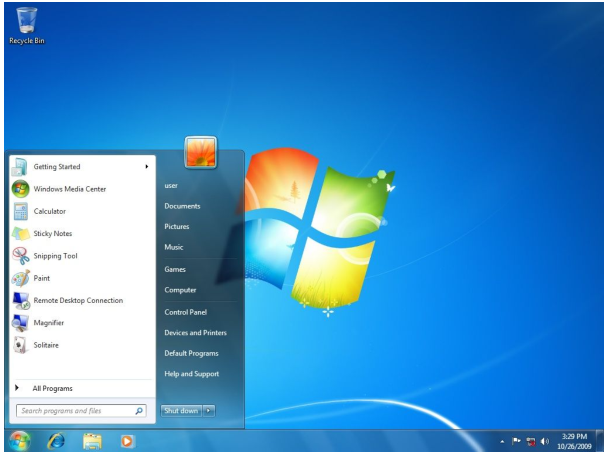
Filter: /opt/IMD/src/5Dwm-1.0/src/*

Examples: Windows Shell

- Windows Shell is the most ubiquitous example of a DE, and its layout and workflow have evolved and changed with Windows itself ever science the DE’s inception in 1995 with the release of Windows 95, which included the now iconic Start menu, taskbar and Windows Explorer.
- Often synonymous with the PC experience for many users due to its market dominance regarding the personal computer.
- Many alternative Linux DE’s are similar to Windows Shell’s design philosophies, such as KDE Plasma’s implementation of a familiar start menu and taskbar.

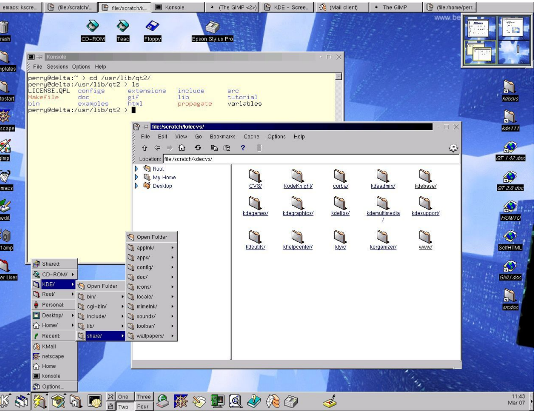




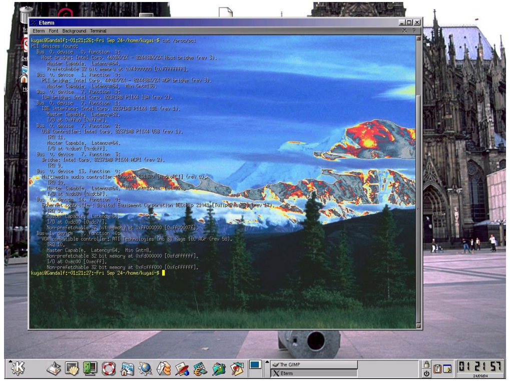


Examples: KDE

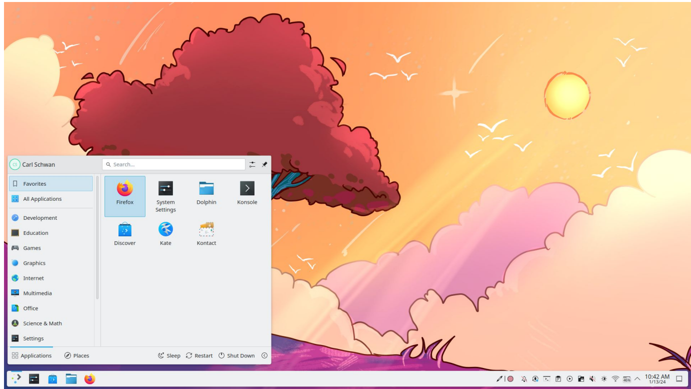
- Established in 1998 as KDE 1.0, built on the Qt framework, provided a full and complete desktop environment with all the bells and whistles you could need in a DE at the time (Window manager, file manager and system tray along with various widgets).
- Known for its modularity and respect to user freedom and customization.
- A modern implementation, known as KDE Plasma, is the DE that the Steam Deck uses.
- Originally known as the “Kool Desktop Environment”.
- Since 2009, KDE now refers to the community of various applications and toolsets, not just the DE.



```
perry@delta:~ > cd /usr/lib/qt2/
perry@delta:/usr/lib/qt2 > ls
LICENSE.QPL  configs      extensions   include      src
Makefile     doc          gif          lib          tutorial
bin          examples     html         propagate    variables
perry@delta:/usr/lib/qt2 >
```

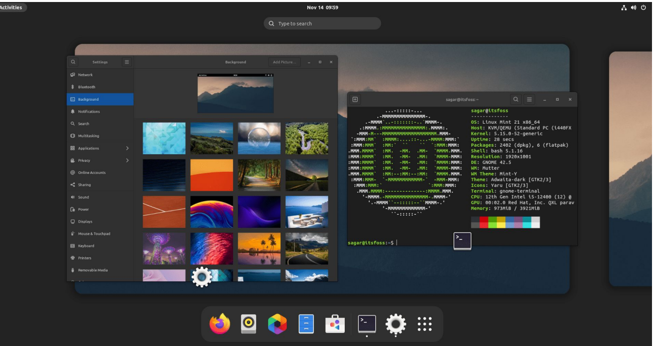


```
kugai@Gandalf:-01:21:26:-Fri Sep 24-/home/kugai-$ cat /proc/pci
PCI devices found:
  Bus  0, device   0, function  0:
    Host bridge: Intel Corp. 440BX/ZX - 82443BX/ZX Host bridge (rev 3).
      Master Capable.  Latency=64.
      Prefetchable 32 bit memory at 0xf4000000 [0xf7ffffff].
  Bus  0, device   1, function  0:
    PCI bridge: Intel Corp. 440BX/ZX - 82443BX/ZX AGP bridge (rev 3).
      Master Capable.  Latency=64.  Min Gnt=136.
  Bus  0, device   7, function  0:
    ISA bridge: Intel Corp. 82371AB PIIX4 ISA (rev 2).
  Bus  0, device   7, function  1:
    IDE interface: Intel Corp. 82371AB PIIX4 IDE (rev 1).
      Master Capable.  Latency=32.
      I/O at 0xffa0 [0xffaf].
  Bus  0, device   7, function  2:
    USB Controller: Intel Corp. 82371AB PIIX4 USB (rev 1).
      IRQ 11.
      Master Capable.  Latency=64.
      I/O at 0xdce0 [0xdcff].
  Bus  0, device   7, function  3:
    Bridge: Intel Corp. 82371AB PIIX4 ACPI (rev 2).
      IRQ 9.
  Bus  0, device  13, function  0:
    Multimedia audio controller: Ensoniq ES1370 [AudioPCI] (rev 0).
      IRQ 10.
      Master Capable.  Latency=64.  Min Gnt=12.Max Lat=128.
      I/O at 0xdc80 [0xdcbf].
  Bus  0, device  14, function  0:
    Ethernet controller: Digital Equipment Corporation DECchip 21041 [Tulip Pass 3] (rev 17).
      IRQ 9.
      Master Capable.  Latency=64.
      I/O at 0xdc00 [0xdc7f].
      Non-prefetchable 32 bit memory at 0xff000000 [0xff00007f].
  Bus  1, device   0, function  0:
    VGA compatible controller: ATI Technologies Inc 3D Rage IIC AGP (rev 58).
      IRQ 10.
      Master Capable.  Latency=64.  Min Gnt=8.
      Non-prefetchable 32 bit memory at 0xfd000000 [0xfdffffff].
      I/O at 0xec00 [0xecff].
      Non-prefetchable 32 bit memory at 0xfcfff000 [0xfcffffff].
kugai@Gandalf:-01:21:27:-Fri Sep 24-/home/kugai-$
```

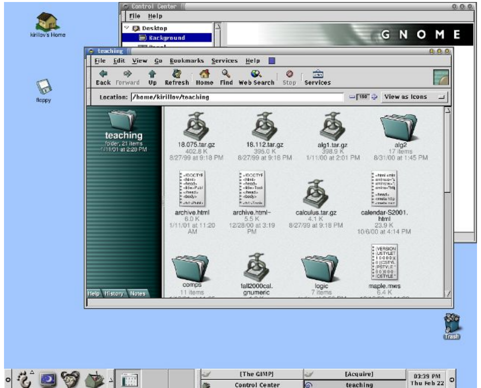


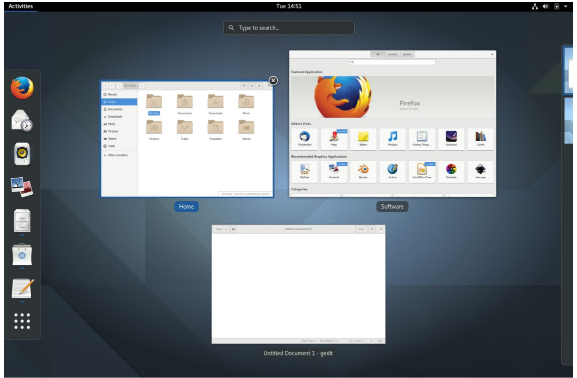
Examples: GNOME

- The first stable version of GNOME (1.0) was released in March 1999, making the project over 24 years old.
- Originally stood for GNU Network Object Model Environment, though this acronym is no longer used officially.
- GNOME was created as a free software alternative to KDE, which at the time used the non-free Qt toolkit as previously mentioned.
- GNOME 3, released in 2011, marked a significant shift in the desktop's design philosophy and user interface, which was initially controversial among some users due to removing customization features and departing from the comfortable desktop-like design along with a focus on touch interfaces, which is reminiscent of Windows 8's DE.
- The DE has now evolved and improved but never truly returned to the traditional desktop design KDE and Windows share.



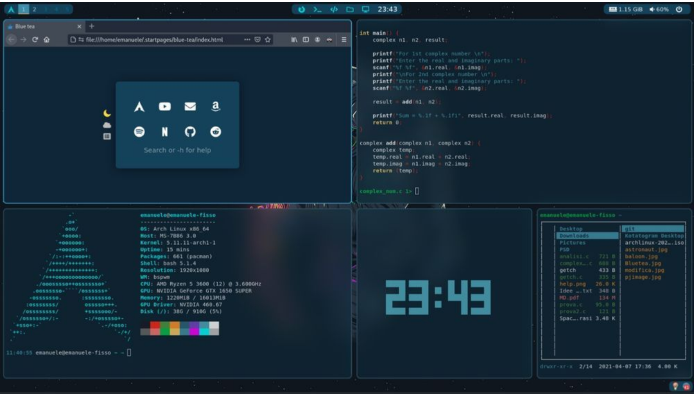
```
sagar@itsfoss
-------------
OS: Linux Mint 21 x86_64
Host: KVM/QEMU (Standard PC (i440FX
Kernel: 5.15.0-52-generic
Uptime: 28 secs
Packages: 2402 (dpkg), 6 (flatpak)
Shell: bash 5.1.16
Resolution: 1920x1001
DE: GNOME 42.5
WM: Mutter
WM Theme: Mint-Y
Theme: Adwaita-dark [GTK2/3]
Icons: Yaru [GTK2/3]
Terminal: gnome-terminal
CPU: 12th Gen Intel i5-12400 (12) @
GPU: 00:02.0 Red Hat, Inc. QXL parav
Memory: 973MiB / 3921MiB

sagar@itsfoss:~$
```





What's a Window Manager?

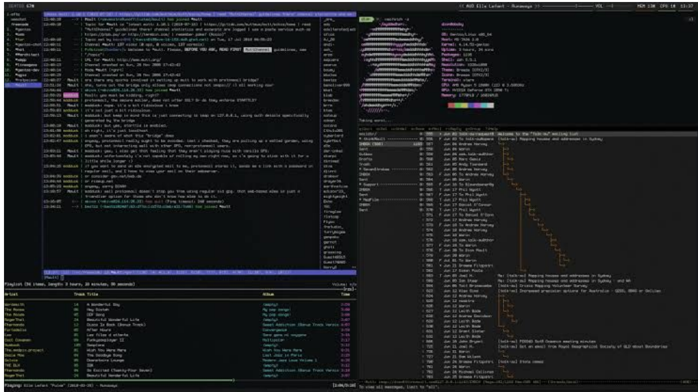
- A window manager (WM) controls the placement, size, appearance and behavior of a window on a screen.
- DE’s have WMs already, with examples being GNOME’s Mutter, and KDE’s KWin.
- Despite that, you do not need a DE to have a WM. Window managers can be used all by themselves.
- This lack of the defining features of a DE makes standalone WM’s feel very minimalistic and focused compared to something like Windows Shell.
- They often garner the most customizability, and are usually utilized with very little mouse input.
- Often “Workspace Oriented”, with various workspaces have different sets of windows.



```
int main() {
    complex n1, n2, result;

    printf("For 1st complex number \n");
    printf("Enter the real and imaginary parts: ");
    scanf("%f %f", &n1.real, &n1.imag);
    printf("\nFor 2nd complex number \n");
    printf("Enter the real and imaginary parts: ");
    scanf("%f %f", &n2.real, &n2.imag);

    result = add(n1, n2);

    printf("Sum = %.1f + %.1fi", result.real, result.imag);
    return 0;
}

complex add(complex n1, complex n2) {
    complex temp;
    temp.real = n1.real + n2.real;
    temp.imag = n1.imag + n2.imag;
    return (temp);
}

complex_num.c 1>
```

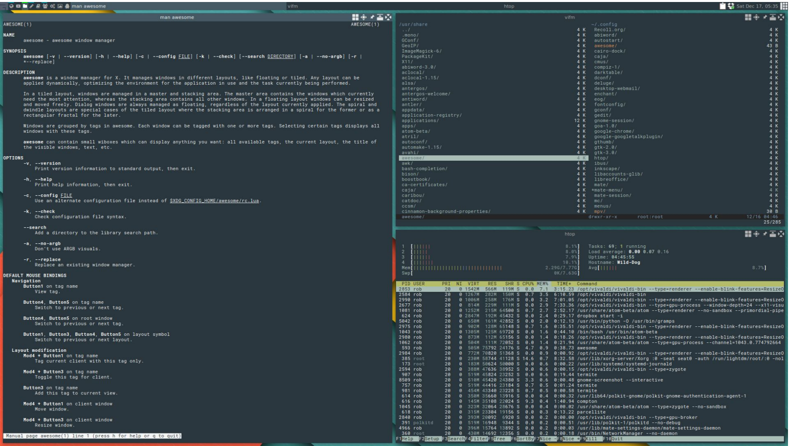
```
emanuele@emanuele-fisso
-----------------------
OS: Arch Linux x86_64
Host: MS-7B86 3.0
Kernel: 5.11.11-arch1-1
Uptime: 15 mins
Packages: 661 (pacman)
Shell: bash 5.1.4
Resolution: 1920x1080
WM: bspwm
CPU: AMD Ryzen 5 3600 (12) @ 3.600GHz
GPU: NVIDIA GeForce GTX 1650 SUPER
Memory: 1220MiB / 16013MiB
GPU Driver: NVIDIA 460.67
Disk (/): 38G / 910G (5%)

11:40:55 emanuele@emanuele-fisso ~ →
```

```
emanuele@emanuele-fisso ~
  Desktop            git
  Downloads          Kotatogram Desktop
  Pictures           archlinux-202….iso
  PSD                astronaut.jpg
  analisi.c   721 B  baloon.jpg
  complex….c  688 B  Bluetea.jpg
  getch       433 B  modifica.jpg
  getch.c     335 B  pjimage.jpg
  help.png   26.0 K
  Idee ….txt  348 B
  MD.pdf      134 M
  prova.c    95.0 B
  prova2.c    121 B
  Spac….rasi 3.48 K
drwxr-xr-x  2/14  2021-04-07 17:36  4.00 K
```



Examples: AwesomeWM

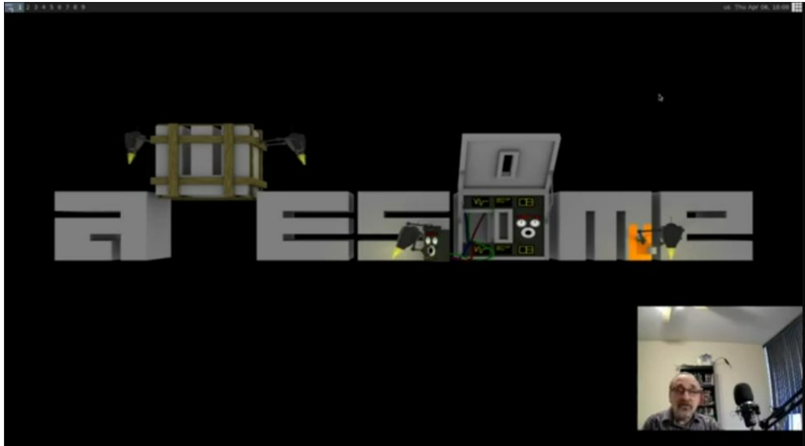
- AwesomeWM started as a fork of the dwm window manager. It was initially nicknamed "jdwm" after the initials of its creator, Julien Danjou.
- It's one of the few window managers that uses Lua for both configuration and extending functionality, making it highly customizable. Almost everything you need can be found in its rc.lua file.
- Has very extensive documentation for all of its features which is rare at times for window managers.
- It's easy to create custom widgets for its built-in status bar (wibar) using simple Lua scripting.



```
AWESOME(1)                                                                              AWESOME(1)

NAME
       awesome - awesome window manager

SYNOPSIS
       awesome [-v | --version] [-h | --help] [-c | --config FILE] [-k | --check] [--search DIRECTORY] [-a | --no-argb] [-r |
       *--replace]

DESCRIPTION
       awesome is a window manager for X. It manages windows in different layouts, like floating or tiled. Any layout can be
       applied dynamically, optimizing the environment for the application in use and the task currently being performed.

       In a tiled layout, windows are managed in a master and stacking area. The master area contains the windows which currently
       need the most attention, whereas the stacking area contains all other windows. In a floating layout windows can be resized
       and moved freely. Dialog windows are always managed as floating, regardless of the layout currently applied. The spiral and
       dwindle layouts are special cases of the tiled layout where the stacking area is arranged in a spiral for the former or as a
       rectangular fractal for the later.

       Windows are grouped by tags in awesome. Each window can be tagged with one or more tags. Selecting certain tags displays all
       windows with these tags.

       awesome can contain small wiboxes which can display anything you want: all available tags, the current layout, the title of
       the visible windows, text, etc.

OPTIONS
       -v, --version
           Print version information to standard output, then exit.

       -h, --help
           Print help information, then exit.

       -c, --config FILE
           Use an alternate configuration file instead of $XDG_CONFIG_HOME/awesome/rc.lua.

       -k, --check
           Check configuration file syntax.

       --search
           Add a directory to the library search path.

       -a, --no-argb
           Don't use ARGB visuals.

       -r, --replace
           Replace an existing window manager.

DEFAULT MOUSE BINDINGS
   Navigation
       Button1 on tag name
           View tag.

       Button4, Button5 on tag name
           Switch to previous or next tag.

       Button4, Button5 on root window
           Switch to previous or next tag.

       Button1, Button3, Button4, Button5 on layout symbol
           Switch to previous or next layout.

   Layout modification
       Mod4 + Button1 on tag name
           Tag current client with this tag only.

       Mod4 + Button3 on tag name
           Toggle this tag for client.

       Button3 on tag name
           Add this tag to current view.

       Mod4 + Button1 on client window
           Move window.

       Mod4 + Button3 on client window
           Resize window.

 Manual page awesome(1) line 1 (press h for help or q to quit)
```

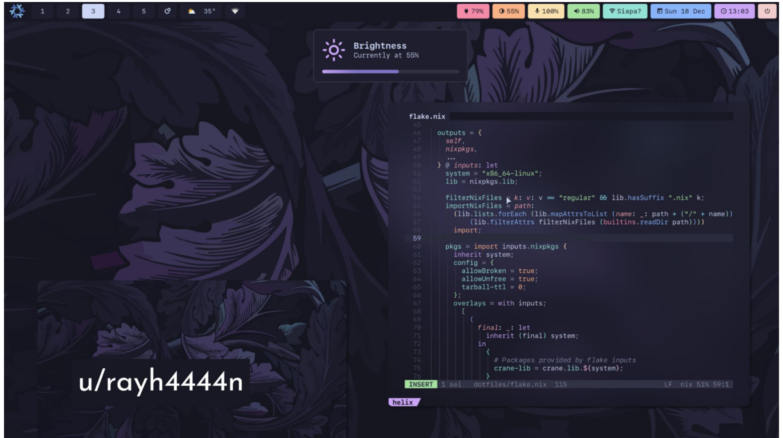
```
/usr/share                                         ~/.config
 ../                                        4 K     Recoll.org/                    4 K
 .mono/                                     4 K     abiword/                       4 K
 GConf/                                     4 K     autostart/                     4 K
 GeoIP/                                     4 K     awesome/                      43 B
 ImageMagick-6/                             4 K     cairo-dock/                    4 K
 PackageKit/                                4 K     caja/                          4 K
 X11/                                       4 K     cmus/                          4 K
 abiword-3.0/                               4 K     compiz-1/                      4 K
 aclocal/                                   4 K     darktable/                     4 K
 aclocal-1.15/                              4 K     dconf/                         4 K
 alsa/                                      4 K     deluge/                        4 K
 antergos/                                  4 K     desktop-webmail/               4 K
 antergos-welcome/                          4 K     enchant/                       4 K
 antiword/                                  4 K     eog/                           4 K
 antler/                                    4 K     fontconfig/                    4 K
 appdata/                                   4 K     gconf/                         4 K
 application-registry/                      4 K     gedit/                         4 K
 applications/                             12 K     gnome-session/                 4 K
 apps/                                      4 K     goa-1.0/                       4 K
 atom-beta/                                 4 K     google-chrome/                 4 K
 atril/                                     4 K     google-googletalkplugin/       4 K
 autoconf/                                  4 K     gthumb/                        4 K
 automake-1.15/                             4 K     gtk-2.0/                       4 K
 avahi/                                     4 K     gtk-3.0/                       4 K
 awesome/                                   4 K     htop/                          4 K
 awk/                                       4 K     ibus/                          4 K
 bash-completion/                           4 K     inkscape/                      4 K
 bison/                                     4 K     libaccounts-glib/              4 K
 boostbook/                                 4 K     libreoffice/                   4 K
 ca-certificates/                           4 K     mate/                          4 K
 caja/                                      4 K    *mate-menu/                     4 K
 caribou/                                   4 K     mate-session/                  4 K
 catdoc/                                    4 K     mc/                            4 K
 ccsm/                                      4 K     menus/                         4 K
 cinnamon-background-properties/            4 K     mpv/                          30 B
 awesome/                                  drwxr-xr-x      root:root        4 K        12/16 04:46
                                                                                             25/285
```

```
  1  [||||||                          8.1%]   Tasks: 69; 1 running
  2  [|||||                           8.0%]   Load average: 0.00 0.07 0.16
  3  [||||||                          7.9%]   Uptime: 04:45:55
  4  [|||||||                        10.1%]   Hostname: Wild-Dog
  Mem[||||||||||||||||||||||||||||||||2.29G/7.77G]   Avg[||||||                     8.7%]
  Swp[                            0K/7.63G]

  PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
 2853 rob        20   0 1542M  566M  119M S  0.0  7.1  3:15.23 /opt/vivaldi/vivaldi-bin --type=renderer --enable-blink-features=ResizeO
 2584 rob        20   0 1267M  282M  150M S  0.7  3.5  6:10.59 /opt/vivaldi/vivaldi-bin
 2990 rob        20   0 1006M  258M  176M S  0.0  3.2  7:01.05 /opt/vivaldi/vivaldi-bin --type=renderer --enable-blink-features=ResizeO
 2677 rob        20   0  814M  229M  111M S  0.0  2.9  7:33.36 /opt/vivaldi/vivaldi-bin --type=gpu-process --window-depth=24 --x11-visu
 1081 rob        20   0 1252M  215M 64500 S  0.7  2.7  2:52.17 /usr/share/atom-beta/atom --type=renderer --no-sandbox --primordial-pipe
  624 rob        20   0 2847M  192M 45432 S  0.0  2.4  0:29.17 dropbox start -i
 5042 rob        20   0  650M  161M 42852 S  0.0  2.0  0:12.13 /usr/bin/python -O /usr/bin/gramps
 2975 rob        20   0  902M  128M 65148 S  0.7  1.6  0:35.51 /opt/vivaldi/vivaldi-bin --type=renderer --enable-blink-features=ResizeO
 1043 rob        20   0 1305M  125M 69720 S  0.0  1.6  0:44.10 /bin/bash /usr/bin/atom-beta
 2980 rob        20   0  873M  112M 65156 S  0.0  1.4  0:18.26 /opt/vivaldi/vivaldi-bin --type=renderer --enable-blink-features=ResizeO
 1062 rob        20   0  504M  111M 72052 S  0.0  1.4  0:21.94 /usr/share/atom-beta/atom --type=gpu-process --channel=1043.0.774792664
  593 rob        20   0  505M 75792 24176 S  4.7  0.9  0:38.73 awesome
 2984 rob        20   0  772M 70820 51368 S  0.0  0.9  0:00.92 /opt/vivaldi/vivaldi-bin --type=renderer --enable-blink-features=ResizeO
  385 root       20   0  230M 58744 41128 S 14.6  0.7  8:32.58 /usr/lib/xorg-server/Xorg :0 -seat seat0 -auth /run/lightdm/root/:0 -nol
  173 root       20   0  183M 50624 50000 S  0.0  0.6  0:00.22 /usr/lib/systemd/systemd-journald
 2594 rob        20   0  388M 47636 38952 S  0.0  0.6  0:00.15 /opt/vivaldi/vivaldi-bin --type=zygote
  907 rob        20   0  519M 45824 23252 S  0.0  0.6  0:19.44 termite
 8509 rob        20   0  510M 45420 24380 S  3.3  0.6  0:00.48 gnome-screenshot --interactive
  757 rob        20   0  519M 44416 23184 S  0.7  0.5  0:01.24 termite
  981 rob        20   0  454M 43340 23228 S  0.7  0.5  0:00.58 termite
  614 rob        20   0  350M 35660 13916 S  0.0  0.4  0:00.32 /usr/lib64/polkit-gnome/polkit-gnome-authentication-agent-1
  616 rob        20   0  145M 35108 22024 S  9.3  0.4  1:40.94 compton
 1045 rob        20   0  323M 32064 28676 S  0.0  0.4  0:00.02 /usr/share/atom-beta/atom --type=zygote --no-sandbox
  618 rob        20   0  315M 23304 19156 S  0.0  0.3  0:13.22 parcellite
 2840 rob        20   0  393M 20092  6920 S  0.0  0.2  0:00.00 /opt/vivaldi/vivaldi-bin --type=gpu-broker
  391 polkitd    20   0  519M 16948  9344 S  0.0  0.2  0:00.51 /usr/lib/polkit-1/polkitd --no-debug
 4966 rob        20   0  396M 15764 13892 S  0.0  0.2  0:00.03 /usr/lib/mate-settings-daemon/mate-settings-daemon
  360 root       20   0  430M 14692 12356 S  0.0  0.2  0:00.18 /usr/bin/NetworkManager --no-daemon
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```



Examples: Hyprland

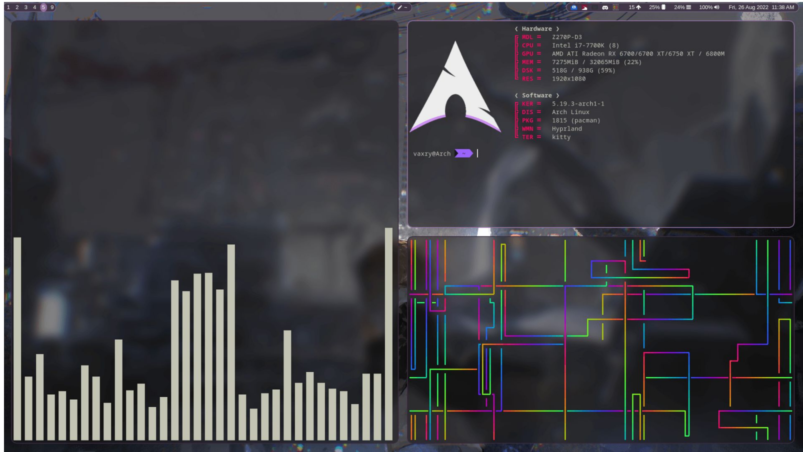
- Hyprland is an up-and-coming WM with an emphasis on functionality that doesn't skimp out on style.
- It's designed for the new Wayland, taking advantage of its modern features and protocols.
- It has built-in animations that can be customized along with the other aspects of its behavior through the hyprland.conf file.
- Has lots of window decoration options such as drop shadows and window dimming.
- Reloads automagically every time hyprland.conf is changed, which is nice for quickly seeing your changes to the settings.



flake.nix

```
  outputs = {
    self,
    nixpkgs,
    ...
  } @ inputs: let
    system = "x86_64-linux";
    lib = nixpkgs.lib;

    filterNixFiles = k: v: v == "regular" && lib.hasSuffix ".nix" k;
    importNixFiles = path:
      (lib.lists.forEach (lib.mapAttrsToList (name: _: path + ("/" + name))
          (lib.filterAttrs filterNixFiles (builtins.readDir path))))
      import;

    pkgs = import inputs.nixpkgs {
      inherit system;
      config = {
        allowBroken = true;
        allowUnfree = true;
        tarball-ttl = 0;
      };
      overlays = with inputs;
        [
          (
            final: _: let
              inherit (final) system;
            in
              {
                # Packages provided by flake inputs
                crane-lib = crane.lib.${system};
              }
```

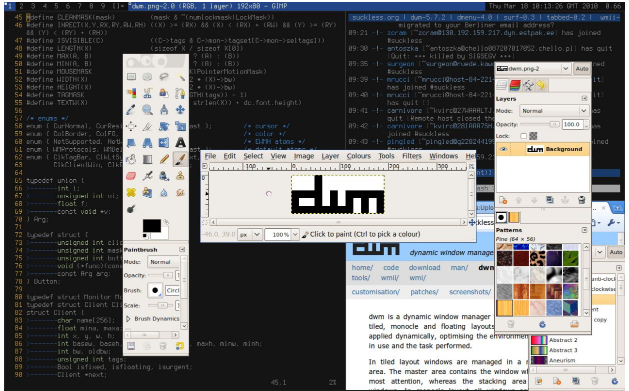
```
INSERT  1 sel   dotfiles/flake.nix  115                    LF  nix 51% 59:1
```



```
❬ Hardware ❭
MDL =   Z270P-D3
CPU =   Intel i7-7700K (8)
GPU =   AMD ATI Radeon RX 6700/6700 XT/6750 XT / 6800M
MEM =   7275MiB / 32065MiB (22%)
DSK =   518G / 938G (59%)
RES =   1920x1080

❬ Software ❭
KER =   5.19.3-arch1-1
DIS =   Arch Linux
PKG =   1815 (pacman)
WMN =   Hyprland
TER =   kitty

vaxry@Arch ~
```

Honorable Mentions

- I3 - Features vi-like control system and an full interprocess communication (IPC) interface which allows it to be customized in various languages.
- Dwm (Dynamic Window Manager) - lacks external dependencies, with its customization done via editing the source code directly in C.
- Bspwm (Binary Space Partitioning Window Manager) - As the name suggests, it uses a binary space partitioning to manage windows, with them being represented as leaves of a binary tree. Configured mostly through shell scripts.



1 2 3 4 5 6 7 8 9 []= dwm.png-2.0 (RGB, 1 layer) 192x80 - GIMP   Thu Mar 18 10:13:26 GMT 2010 0.66

```
#define CLEANMASK(mask)          (mask & ~(numlockmask|LockMask))
#define INRECT(X,Y,RX,RY,RW,RH) ((X) >= (RX) && (X) < (RX) + (RW) && (Y) >= (RY)
&& (Y) < (RY) + (RH))
#define ISVISIBLE(C)             ((C->tags & C->mon->tagset[C->mon->seltags]))
#define LENGTH(X)                (sizeof X / sizeof X[0])
#define MAX(A, B)                           ? (A) : (B))
#define MIN(A, B)                           ? (A) : (B))
#define MOUSEMASK                          K|PointerMotionMask)
#define WIDTH(X)                           2 * (X)->bw)
#define HEIGHT(X)                          2 * (X)->bw)
#define TAGMASK                            GTH(tags)) - 1)
#define TEXTW(X)                            strlen(X)) + dc.font.height)

/* enums */
enum { CurNormal, CurResi                  ast };          /* cursor */
enum { ColBorder, ColFG,                                   /* color */
enum { NetSupported, NetW                                  /* EWMH atoms */
enum { WMProtocols, WMDel                  ast };          /* default atoms */
enum { ClkTagBar, ClkLtSy                  xt,
       ClkClientWin, ClkR                  ;

typedef union {
        int i;
        unsigned int ui;
        float f;
        const void *v;
} Arg;

typedef struct {
        unsigned int clic
        unsigned int mask
        unsigned int butt
        void (*func)(cons
        const Arg arg;
} Button;

typedef struct Monitor Mo
typedef struct Client Cli
struct Client {
        char name[256];
        float mina, maxa;
        int x, y, w, h;
        int basew, baseh,                      , maxh, minw, minh;
        int bw, oldbw;
        unsigned int tags;
        Bool isfixed, isfloating, isurgent;
        Client *next;
                                                          45,1          2%
```

```
suckless.org | dwm-5.7.2 | dmenu-4.0 | surf-0.3 | tabbed-0.2 |  wmii-
           migrated to your Berliner email address?
09:21 -!- zcram [~zcram@130.192.159.217.dyn.estpak.ee] has joined
          #suckless
09:30 -!- antoszka [~antoszka@chello087207017052.chello.pl] has quit
          [Quit: +++ killed by SIGSEGV +++]
09:35 -!- surgeon [~surgeon@ruede.kaw                          ned
          #suckless
09:39 -!- mrucci [~mrucci@host-84-221-                         it]
          has joined #suckless
09:40 -!- mrucci [~mrucci@host-84-221-                         it]
          has quit []
09:41 -!- carnivore [~kvirc@27WAAALTJ.                         as
          quit [Remote host closed the
09:42 -!- carnivore [~kvirc@28IAAA7SH.                         as
          joined #suckless
09:43 -!- pingled [~pingled@g22824419                          ined
```

-46.0, 39.0 px 100 % Click to paint (Ctrl to pick a colour)

Paintbrush
Mode: Normal
Opacity:
Brush: Circl
Scale:
Brush Dynamics

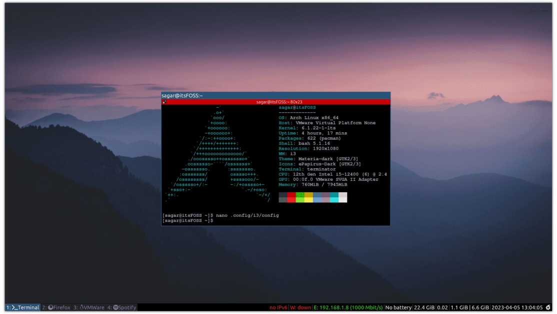
dwm.png-2 Auto
Layers
Mode: Normal
Opacity: 100.0
Lock:
Background
Patterns
Pine (64 × 56)

dwm dynamic window manage
home/ code download man/ dwm
tools/ wmii/ wmi/
customisation/ patches/ screenshots/

dwm is a dynamic window manager
tiled, monocle and floating layouts
applied dynamically, optimising the environmen
in use and the task performed.

In tiled layout windows are managed in a
area. The master area contains the window wh
most attention, whereas the stacking area

File Edit Select View Image Layer Colours Tools Filters Windows Help
anti-clock
clockwise
copy
Abstract 2
Abstract 3
Aneurism

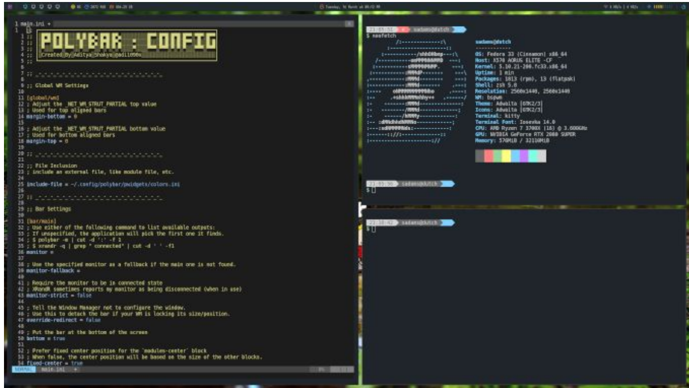


sagar@itsFOSS:~

sagar@itsFOSS:~ 80x23

```
                   -`                    sagar@itsFOSS
                  .o+`                   -------------
                 `ooo/                   OS: Arch Linux x86_64
                `+oooo:                  Host: VMware Virtual Platform None
               `+oooooo:                 Kernel: 6.1.22-1-lts
               -+oooooo+:                Uptime: 4 hours, 17 mins
             `/:-:++oooo+:               Packages: 622 (pacman)
            `/++++/+++++++:              Shell: bash 5.1.16
           `/++++++++++++++:             Resolution: 1920x1080
          `/+++ooooooooooooo/`           WM: i3
         ./ooosssso++osssssso+`          Theme: Materia-dark [GTK2/3]
        .oossssso-````/ossssss+`         Icons: ePapirus-Dark [GTK2/3]
       -osssssso.      :ssssssso.        Terminal: terminator
      :osssssss/        osssso+++.       CPU: 12th Gen Intel i5-12400 (6) @ 2.4
     /ossssssss/        +ssssooo/-       GPU: 00:0f.0 VMware SVGA II Adapter
   `/ossssso+/:-        -:/+osssso+-     Memory: 760MiB / 7945MiB
  `+sso+:-`                 `.-/+oso:
 `++:.                           `-/+/
 .`                                 `/

[sagar@itsFOSS ~]$ nano .config/i3/config
[sagar@itsFOSS ~]$
```

1:>_Terminal 2:Firefox 3:VMWare 4:Spotify   no IPv6 | W: down | E: 192.168.1.8 (1000 Mbit/s) | No battery | 22.4 GiB | 0.02 | 1.1 GiB | 6.6 GiB | 2023-04-05 13:04:05



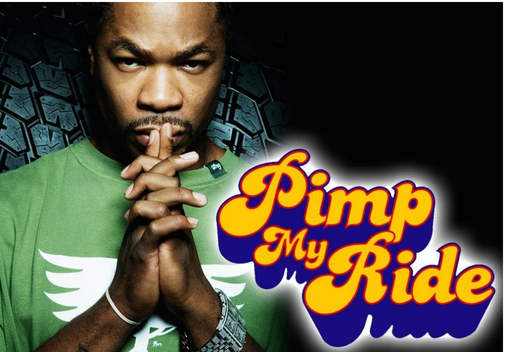
main.ini +

```
;;
;;  POLYBAR : CONFIG
;;  Created By Aditya Shakya @adi1090x
;;
;;

;; _-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_

;; Global WM Settings

[global/wm]
; Adjust the _NET_WM_STRUT_PARTIAL top value
; Used for top aligned bars
margin-bottom = 0

; Adjust the _NET_WM_STRUT_PARTIAL bottom value
; Used for bottom aligned bars
margin-top = 0

;; _-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_

;; File Inclusion
; include an external file, like module file, etc.

include-file = ~/.config/polybar/pwidgets/colors.ini

;; _-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_

;; Bar Settings

[bar/main]
; Use either of the following command to list available outputs:
; If unspecified, the application will pick the first one it finds.
; $ polybar -m | cut -d ':' -f 1
; $ xrandr -q | grep " connected" | cut -d ' ' -f1
monitor =

; Use the specified monitor as a fallback if the main one is not found.
monitor-fallback =

; Require the monitor to be in connected state
; XRandR sometimes reports my monitor as being disconnected (when in use)
monitor-strict = false

; Tell the Window Manager not to configure the window.
; Use this to detach the bar if your WM is locking its size/position.
override-redirect = false

; Put the bar at the bottom of the screen
bottom = true

; Prefer fixed center position for the `modules-center` block
; When false, the center position will be based on the size of the other blocks.
fixed-center = true
```

```
$ neofetch
                                   sadams@dutch
                                   ------------
                                   OS: Fedora 33 (Cinnamon) x86_64
                                   Host: X570 AORUS ELITE -CF
                                   Kernel: 5.10.21-200.fc33.x86_64
                                   Uptime: 1 min
                                   Packages: 1813 (rpm), 13 (flatpak)
                                   Shell: zsh 5.8
                                   Resolution: 2560x1440, 2560x1440
                                   WM: bspwm
                                   Theme: Adwaita [GTK2/3]
                                   Icons: Adwaita [GTK2/3]
                                   Terminal: kitty
                                   Terminal Font: Iosevka 14.0
                                   CPU: AMD Ryzen 7 3700X (16) @ 3.600GHz
                                   GPU: NVIDIA GeForce RTX 2080 SUPER
                                   Memory: 570MiB / 32110MiB

$
$
```

Ricing

- Ricing is simply a term for when you personalize and customize your desktop environment / window manager..
- The term "rice" is derived from "Race Inspired Cosmetic Enhancements," which refers to the modifications made to cars to improve their appearance rather than their performance.
- DE’s usually include means to customize their appearance, so this section will be more about window managers than desktop environments.
- The process is usually different for each WM, so I will mostly cover how to start ricing AwesomeWM and Hyprland.



How 2 Pimp 101: AwesomeWM

- Ricing in Awesome boils down to two things (mostly), your rc.lua file and your theme.lua file, which is usually in a folder with the assets that theme needs. The theme.lua controls the colors and appearance, and rc.lua controls the layout and control.
- Start by making a directory in .config called awesome, and the copy your default rc.lua and your default theme.lua file from its location in /usr/share/awesome
- Then you may bust out a text editor of your choice and you can begin customization.
- A good place to learn and experiment with configs is to go to https://awesomewm.org and navigate to the recipes section. The set of simple widgets and awesome-copycats sections are good to help add premade widgets and see how other people configure their system.
- I also have a demo from my personal laptop you may view!

How 2 Pimp 102: Hyprland

- Hyprland is slightly skeletal, so it may need additional applications in order to get it to how you want it.
- After downloading the package through your package manager, you'll want to navigate to ~/.config/hypr/hyprland.conf to begin tailoring it to your needs.
- In the config file, every configuration option is denoted with the following format : COMMAND=VALUE, with comments being denoted with # and sections denoted with *<nameofsection> {<contentsofsection>}.*
- What might not be immediately apparent is that the default config file does not have every configuration option listed, so you'll need to refer to https://wiki.hyprland.org/configuring/configuring-hyprland/ for the additional options.
- You won't have a built in bar like you do in AwesomeWM, so you'll need to find and make your own. I recommend Waybar(https://github.com/Alexays/Waybar) due to how easy it is to configure, with most of its visual elements being in css.
- https://hyprland.org/ also has a few examples you can look at which can really help you understand all that goes on in making a good rice.
- I also have an example demo of Hyprland!