

Linux Tiling Window Managers with ooRexx

The 33rd International Rexx Symposium

Agenda



Concept,
History,
Comparison

Open Source Jungle,
Distributions

Configuration

Files,
Pango Markup,
ADDRESS,

Hooks & Signal,
Mouse Input,
Menus

Briefly about me:



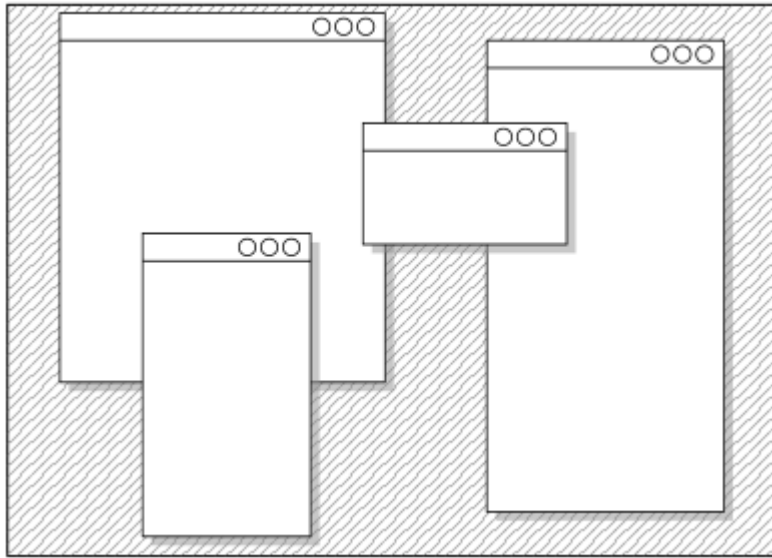
- Background in Psychology
 - R for statistical analysis
- Linux user
 - Bash (never been productive with it)
- Several failed attempts to be productive with programming languages
 - Blitz Basic (School), Pascal/Delphi (School), Java (University), ...
- ooRexx since 2020/2021:
 - **for research project:** scraping information from the web, generating experimental setups (html/xml), reformatting data, ...
 - tinkering with Linux



Window Manager Types

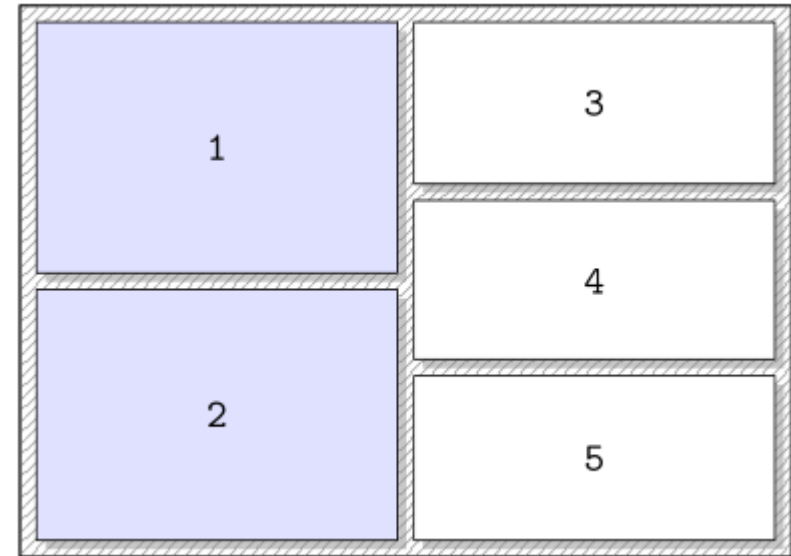


Stacking



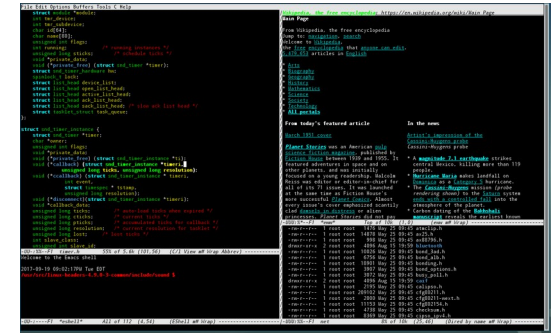
- MS Windows, MacOS, Linux (Gnome, Xfce, KDE Plasma, ...)

Tiling

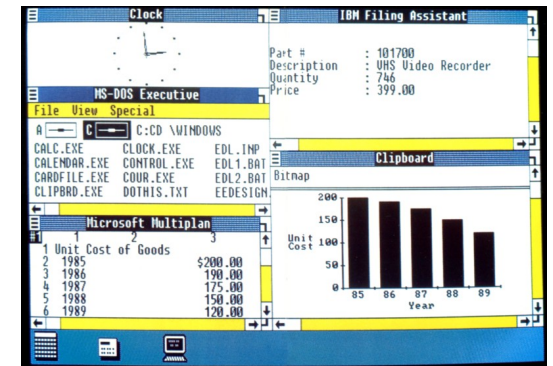


- 25+ WMs for **X Window Systems (X11)** and **Wayland**

- **1970s:** Emacs as the first tiling application
- **1981:** Xerox Star with first tiling window manager (WM)
- **1985:** Due to a lost lawsuit with Apple, Windows 1.0 had to use tiles, later versions abandoned this concept in favour of stacking.
 - Windows 8 failed attempt to reintroduce tiles (2012)
 - Windows 10 has Snap Assist and Fancy Zones (2015)
- **1988:** Siemens RTL WM is the first tiling window manager for X window systems (X11)
- **2000s:** rise of open-source tiling WM for X11
- **2015:** MacOS X 10.11 introduces tiling of two applications.



Emacs

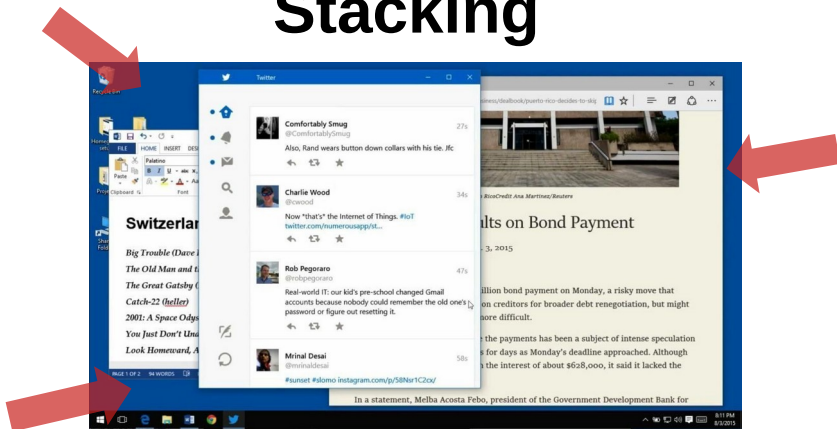


Windows 1.0

A Quick Comparison

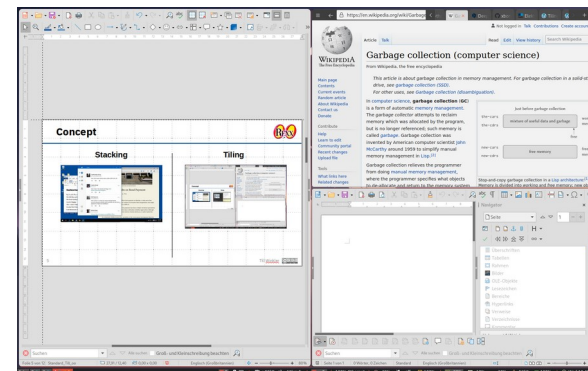


Stacking



- **Mouse-driven interaction**
 - easy to learn
- **Wastes space on the display**
 - tile bar
 - empty space

Tiling

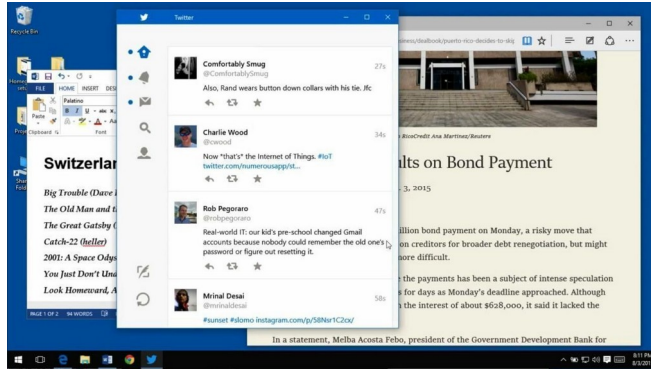


- **Keyboard-driven interaction**
 - hard to learn
 - left-/mixed-handed and Parkinson's patients
- **Saves space on the display**
 - no title bar
 - no empty space

A Quick Comparison

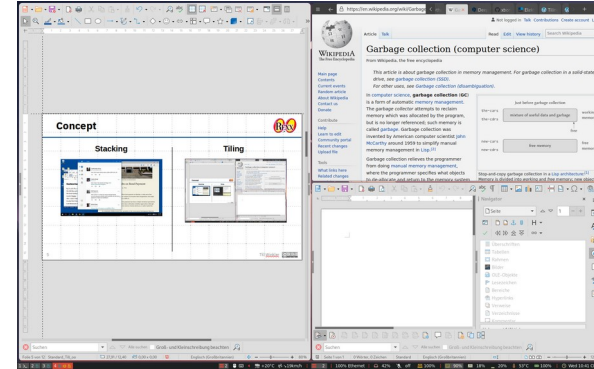


Stacking



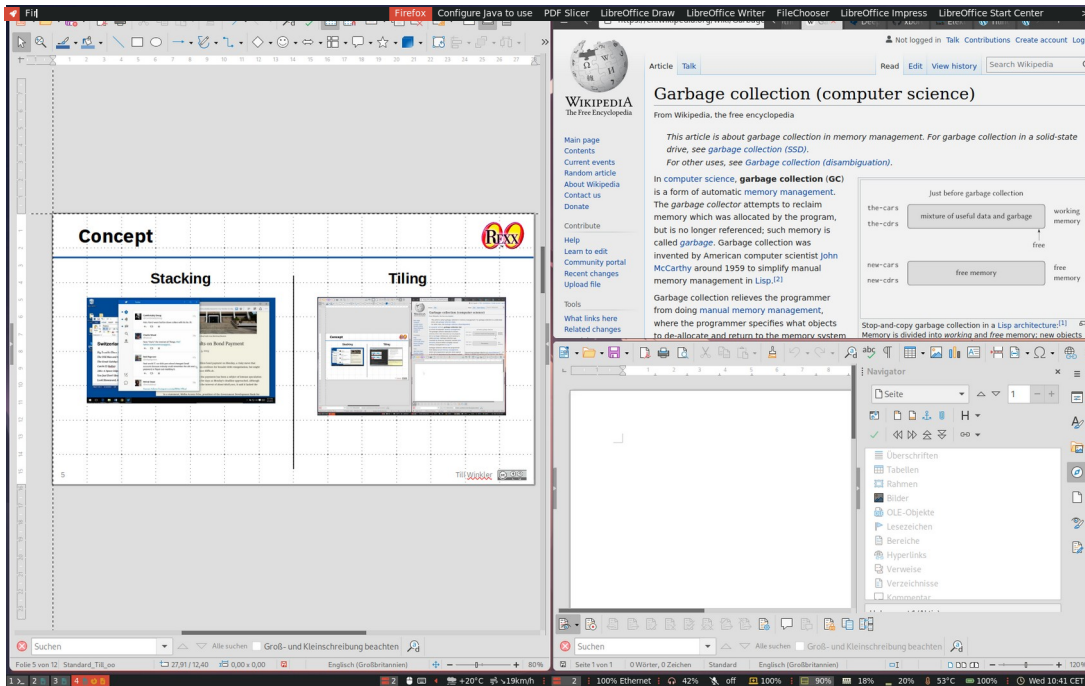
- **Less Productive**
 - resizing
 - re-arranging/moving windows
 - searching windows
- **Higher Resource Consumption**

Tiling



- **More Productive**
 - no resizing
 - limited arranging/moving windows
 - good for multi-monitor setups
- **Lower Resource Consumption**

Components that make up a complete desktop environment

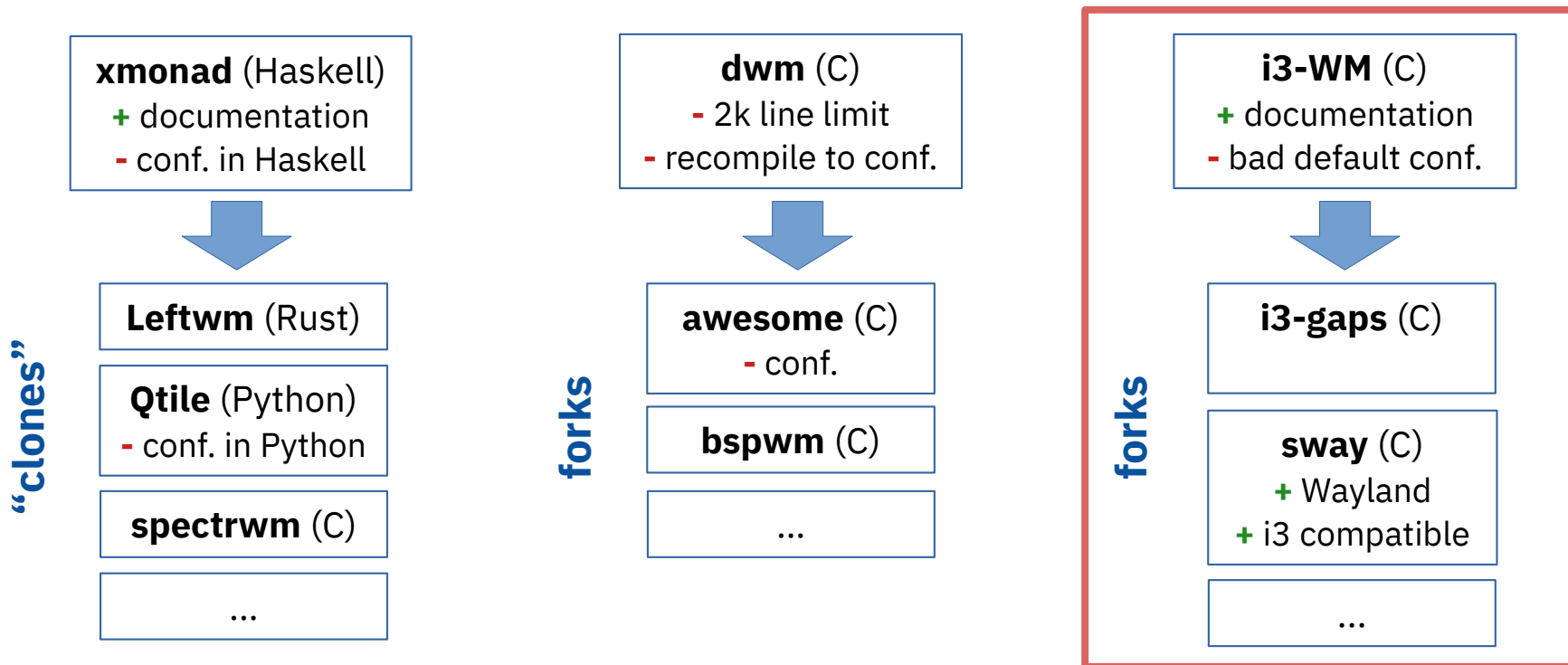


- **Window Manager**
- **Compositor** (optional)
 - visual effects
- **Status-Bar**
- **Application Launcher**
 - keyboard-driven menu

workspace-indicator

Status-Bar with function blocks

Open Source Jungle: 25+ Options



- **Honorable mentions:** FrankenWM (C), Herbstluf (C++), Notion (C, Lua), Stumpwm (Lisp), Worm (Nim), ...
- **Overview:** <https://wiki.archlinux.org/title/Comparison_of_tiling_window_managers> (09.09.2022)

Open Source Jungle: Components for i3-WM

- **Status Bar**

- i3status, pyi3status, i3bar, candybar, lemonbar, tint2, polybar, excalibar, ...
- **i3blocks**: <<https://github.com/vivien/i3blocks>> (09.09.2022)

- **Application Launcher**

- bashrun2, gmrn, xrunner, pmenu, lighthouse, launch, rofi, ...
- **dmenu**: <<https://tools.suckless.org/dmenu/>> (09.09.2022)

- **Compositor**

- compton, Xcompmgr, ...
- **picom**: <<https://github.com/yshui/picom>> (09.09.2022)



- **Ubuntu-Regolith** (Debian branch) <<https://regolith-desktop.com/>>

- **missing:** ooRexx 5.0 (download), dmenu, i3blocks

```
sudo dpkg -i <package name>.deb
sudo apt-get install dmenu i3blocks
```

- **Fedora i3** (Red Hat branch) <<https://spins.fedoraproject.org/en/i3/>>

- **missing:** ooRexx 5.0 (download), i3blocks, picom

```
rpm -i <package name>.rpm
sudo dnf install i3blocks picom
```

- **Manjaro i3** (Arch Linux branch) <<https://manjaro.org/download/>>

- **missing:** ooRexx 5.0 (build), i3blocks

```
yay -S oorexx-beta
yay -S i3blocks
```



5.0.0.r12469-1 is present in arch user repository

- builds using [cmake](#)
- **Maintainer:** sardo

Minimal Configuration: i3 and i3blocks

- i3-WM: <<https://i3wm.org/docs/userguide.html>> (09.09.2022)
- i3blocks: <<https://vivien.github.io/i3blocks/>> (09.09.2022)

i3-WM: ~/.config/i3/config

```
...
bar {
  id          bar-1
  output      primary
  status_command i3blocks -c ~/.config/i3/i3blocks
  position    bottom
  mode        dock

  colors      {
    background #1d2121
    ...
  }
}
...
```

i3blocks: ~/.config/i3/i3blocks

```
markup=pango

[first_block]
interval=10
command=rexx ~/Rexx/first_block.rexx
SYMBOLS=🕒 🕒 🕒 🕒 🕒

[second_block]
interval=once
signal=16
command=rexx ~/Rexx/second_block.rexx
SPEAKER_HEAD=🔊 🔊
```


Everything is a file, 1

- Key defining feature of Unix and its derivatives
- Wide range of input/output resources are simple streams of bytes, accessible via virtual filesystems.
 - **Advantage:** Same set of simple tools can be used for a wide range of resources.
- **Process related:** `/proc/...`
 - runtime information provided by the kernel, e.g.:
 - `/proc/PID/stat` Process status.
 - `/proc/PID/mem` Memory held by this process.
- **Non-process related:** `/sys/...` (or: `/proc/sys/...`)
 - additional information about kernel subsystems, e.g.:
 - `/sys/class/thermal/...` Information related to the thermal status.

Everything is a file, 2



- Reading: `/sys/class/thermal/thermal_zone0/temp`

```
/* [first_block] */  
ZoneTemp = .Stream~new("/sys/class/thermal/thermal_zone0/temp")~LineIn(1)  
Temp = ZoneTemp/1000~round  
  
Symbols = VALUE("SYMBOLS", "ENVIRONMENT")  
  
if Temp <= 50 then Pos = 1  
  else Pos = 5  
  
UseSymbol = symbols~word(Pos)  
  
say UseSymbol Temp || "°C"
```

Output:

```
1 >_>_ 2 >_>_ 3 76 4 76 76°C
```



Pango Markup, 1

- Simple markup language that allows to specify attributes to change the appearance of the output text.
 - **Attributes:** `font_family`, `size`, `style`, `weight`, `stretch`, `foreground`, `background`, `underline`, ...
- **Example:**

```
<span foreground="#ffc000"> Text will be red! </span>
```
- Pango Markup forms the core of text and font handling for GTK.
 - Standard for desktop environments such as Gnome, Xfce, LXDE, Cinnamon, ...
 - Link: <https://docs.gtk.org/Pango/> (09.09.2022)



- Change output appearance with Pango Markup.

```

/* [first_block] */
ZoneTemp = .Stream~new("/sys/class/thermal/thermal_zone0/temp")~LineIn
Temp = ZoneTemp/1000~round

Symbols = VALUE("SYMBOLS",,"ENVIRONMENT")

if Temp <= 50 then Pos = 1
  else Pos = 5

UseSymbol = symbols~word(Pos)

Select case Pos
  when 1 then FgColor = "#ff6666"
  when 5 then FgColor = "#ff3232"
  otherwise FgColor = "#f7f7f7"
end
ColorSymbol = '<span foreground='FgColor'> 'UseSymbol' </span>'

say ColorSymbol Temp || "°C"

```

Output:

```

1 >_>_ 2 >_>_ 3 6 4 6
🌡️ 80°C

```



- Using **ADDRESS** to gain information or control command-line programs.
 - `pactl list sinks` Information on PipeWire Multimedia Server
 - `pactl -- set-sink-volume 0 80%` Set sink (audio output) volume to 80%
 - `pacmd list-sinks` Information on PulseAudio Sound Server

```
/* [second_block] */
command="pactl list sinks"  -- only for pipewire
```

```
SinksArr = .array~new
ADDRESS System command with output using (SinksArr)
```

```
do name over SinksArr
  if name~word(1) = "Volume:" then CurrentVolume = name~word(5)
end
```

```
say CurrentVolume
```



```
Sink #47
... cut ...
Mute: no
Volume: front-left: 27295 / 42% ...
      balance 0,00
... cut ...
Active Port: analog-output-speaker
```

Output:


```
1 > _ > _ > _ 2 > _ 3 b 4 b 0 24°C | 42%
```



Hooks and Real Time Signals, 1

- **Hooks** enable to alter/augment behaviour by intercepting and reacting to *function calls*, *messages* or **events**.
 - Which events are available and how the necessary scripts look like depends strongly on the distribution and the installed packages.
- In Linux many hooks can be added to `/etc/...`
 - `/etc/acpi/events/...` Events related to power management.
 - `/etc/cron.daily/...` Daily cronjob events.
 - `/etc/udev/rules.d/...` Events related to attached USB devices.
- **ACPI-Example:**
 - `/etc/acpi/events/jack`

```
event=jack/headphone
action=<do something>
```

Hooks and Real-Time Signals, 2

- **Signals** are notifications delivered to a process by the kernel.
 - Standard signals have been around since the early days of Unix, while real-time signals were specified later (IEEE Std 1003.1b-1993).
- In contrast to standard signals, **real-time signals** are supplied in ascending order and are delivered even if the same signal is pending several times.
- **ACPI-Example with Real-Time Signal:**
 - `pkill` is a utility for sending signals present in all Linux distributions (`procps` package).
 - `/etc/acpi/events/jack`

```
event=jack/headphone  
action=pkill -RTMIN+16 i3blocks
```

Hooks and Real Time Signals, 3



Headphones
plugged: **in** or **out**



`/etc/acpi/events/jack`

```
event=jack/headphone
action=pkill -RTMIN+16 i3blocks
```



`~/.config/i3/i3blocks`

```
[second_block]
interval=once
signal=16
command=rexx ~/.Rexx/second_block.rexx
SPEAKER_HEAD=🔊🔇
```

```
/* [second_block] */
command="pactl list sinks" -- only for pipewire

SinksArr = .array~new
ADDRESS System command with output using (SinksArr)

ItemNumber = SinksArr~items - 2

Symbols = VALUE("SPEAKER_HEAD", "ENVIRONMENT")
if SinksArr[ItemNumber]~word(3) = "analog-output-speaker" then Pos = 1
else Pos = 2

UseSymbol = symbols~word(Pos)
say UseSymbol
```



```
Sink #47
... cut ...
Mute: no
Volume: front-left: 27295 / 42% ...
        balance 0,00
... cut ...
Active Port: analog-output-headphones
```

Output:

```
1 >_>_>_ 2 >_ 3 🔊 4 🔇 30°C 🔊
```



Mouse Input



- Mouse input is provided as environmental variable.

```
Select case VALUE("BLOCK_BUTTON", "ENVIRONMENT")
  when 1 then say "Clicked Left"
  when 2 then say "Clicked Middle"
  when 3 then say "Clicked Right"
  when 4 then say "Scrolled Up"
  when 5 then say "Scrolled Down"
  otherwise say "Provide Input"
end
```

Output:

```
1 >_>_>_ 2 >_ 3 4 26°C | Provide Input
```

Output (if clicked left):

```
1 >_>_>_ 2 >_ 3 4 26°C | Clicked Left
```

Output (if scrolled up):

```
1 >_>_>_ 2 >_ 3 4 26°C | Scrolled Up
```

Simple Menu using Dmenu, 1

- **Dmenu** is a lightweight and keyboard-driven menu for X window systems.
 - Reads **text** from **stdin**, and creates a menu with one item for each line.

```
echo -e "item1\nitem2\nitem3" | dmenu
```



```
| item1 item2 item3
```

- Selected line is printed to **stdout**.

```
item2
```

- Other application launchers (e.g. rofi, lighthouse, ...) work the same way.

Simple Menu using Dmenu, 2

- Simple Dmenu with Glyphs

```
DmenuOut = .array~of("🐧 Linux", "🍏 Apple", "🪟 Windows")

UserChoice = .array~new
ADDRESS System 'echo -e' "'DmenuOut'" ' | dmenu' with output using (UserChoice)

Select Case UserChoice[1]
  when "🪟 Linux" then say "User has chosen Linux."
  when "🍏 Apple" then say "User has chosen Apple."
  when "🪟 Windows" then say "User has chosen Windows."
  otherwise Nop
end
```

A screenshot of a terminal window showing a dmenu menu. The menu has a black background and three items: "🐧 Linux" (with a white cursor bar to its left), "🍏 Apple" (highlighted in red), and "🪟 Windows".

```
| 🐧 Linux 🍏 Apple 🪟 Windows
```

Output:

```
User has chosen Apple.
```

Roundup: Nutshell Examples

What can be achieved?



← Scroll →

1/6
1/5
BTC: €21306 +1.55% (1d)
off
increase

decrease
00:00:00
Sun 11-09-22
00:00:00

+19°C
11km/h
1
71% UPC3310018
104%
off
52%
94%
23%
3%
54°C
97%
Sun 14:55 CET

cUrl *controls* *network* *audio* *display* *basic stats* *time*

Left

forecast, details *update, mount, ...* *connect* *mute* *presentation* *htop, powertop, ...* *ntp service, calendar, ...*

Right

94% | File Manager Details ...

0 8 16 24 32 40 48 56 64 72 80 88 96 [104] 112 120

Block Install Remove Details History Downgrade Lostfiles Edit

Vienna: +19°C | Forecast Block Location Edit

- Classes and Methods for handling: *symbols, colors, scrolling, dmenu* and *other tasks*



Roundup: Tiling Window Managers



- Good opportunity for students to get to know **Linux** and **ooRexx**
 - “Simple” compared to other desktop environments such as Gnome, Xfce or Plasma
 - Pretty results with practical utility
- Interesting keyboard-driven WM concept
 - Saves display space and can boost productivity
 - Fairly low resource consumption – Pseudo comparison (idle, 1 CPU):

	Fedora i3	Ubuntu Regolith	Ubuntu Gnome	Manjaro XFCE
Main Memory	335MB	392MB	750MB	560MB
Load Average				
• 1 min	0.01	0.05	0.12	0.04

Thank You!

till.winkler@wu.ac.at