

# A Memory Comparison of Light Linux Desktops – Part 3

Source:

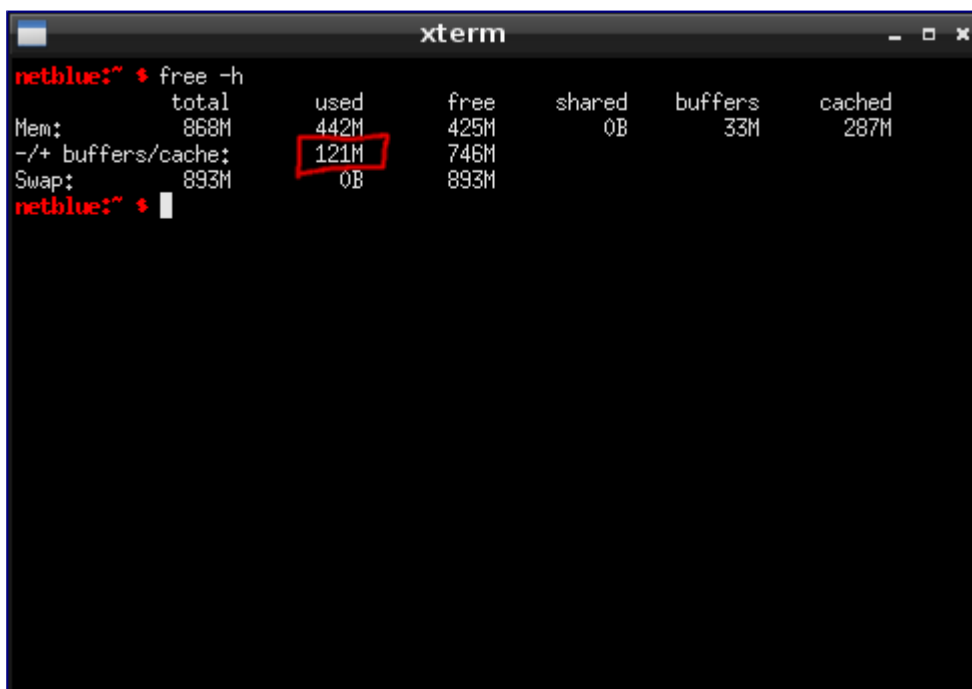
<https://l3net.wordpress.com/2014/02/15/a-memory-comparison-of-light-linux-desktops-part-3/>

Linux kernel manages all RAM memory in your computer. Unused memory goes into a special buffering pool, where the kernel caches all recently used data. If a process attempts to read a file and the kernel already has the file cached, reading it is as fast as reading RAM.

Filesystem-heavy task, such as compiling source code, processing video files, etc. benefit from as much free memory as possible in buffering pool. It is not uncommon today to see users with powerful systems running [tiling window managers](#) in only a few megabytes of memory. Also, with the personal computer market in decline, people tend to keep their computers longer.

In this article I continue the measurements started in [part 1](#) and [part 2](#) of this series.

I use `free` command to measure memory. It basically prints out values provided by the kernel. Of interest to us is the number on `-/+ buffers/cache` line, 121MB in the example below:



```
netblue:~$ free -h
              total        used        free      shared    buffers     cached
Mem:           868M          442M          425M           0B           33M          287M
-/+ buffers/cache: 121M          746M          893M
Swap:          893M           0B          893M
```

## Measuring desktop memory

I measure the memory before I start the window manager (WM) or desktop environment (DE). I do another measurement after, and I have the amount of RAM required to run the desktop. It is an idle desktop with only the default tasks envisioned by the developers. Everything runs in a LXC container set up with [virtenv](#). The computer I am using is a 64bit Debian 7 system, with WMs and DEs as distributed by Debian stable.

## TinyWM

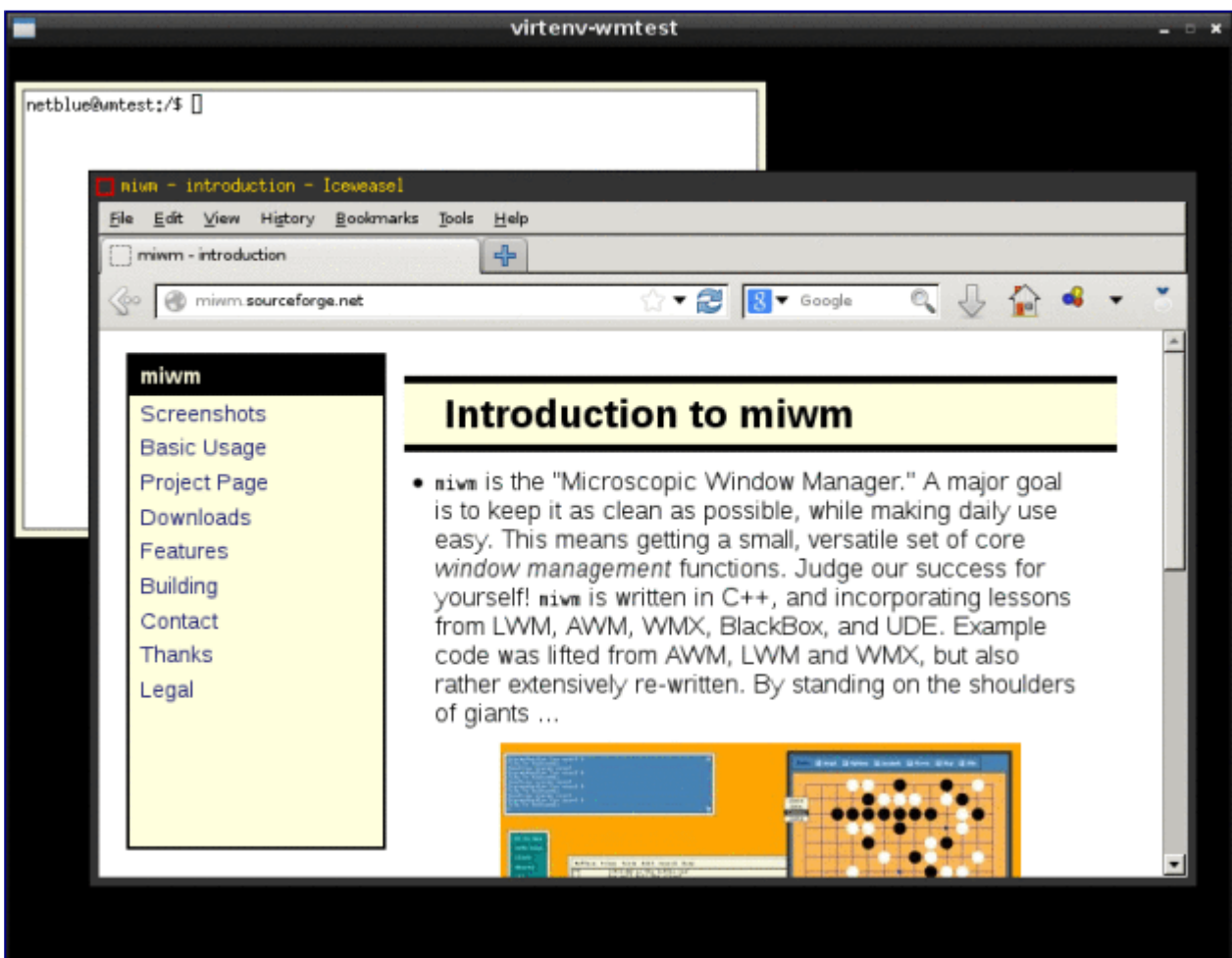
If you are into “less memory is better”, you definitely need to check out [TinyWM](#). Written by Nick Welch in about 50 lines of C code, TinyWM is a “ridiculously tiny window manager”, used to serve as a basic WM programming example. You might also want to check on developer’s webpage the Python version, Lisp version, and Ada version.

TinyWM will move, resize, focus, and raise your windows. There are no window decorations, it looks just like regular X. On Debian/Ubuntu install TinyWM as `apt-get install tinywm`. Start it as `tinywm`. It runs in less than 0.2MB of memory.

## miwm

From tiny we move to microscopic. [miwm](#) (Microscopic Window Manager) implements a small, versatile set of core window management functions, with the goal of keeping the window manager as clean as possible. You would need to supply your own launcher, taskbar and menus using external programs.

On Debian/Ubuntu install miwm as `apt-get install miwm`. Start it as `miwm`. It runs in 0.5MB of memory.

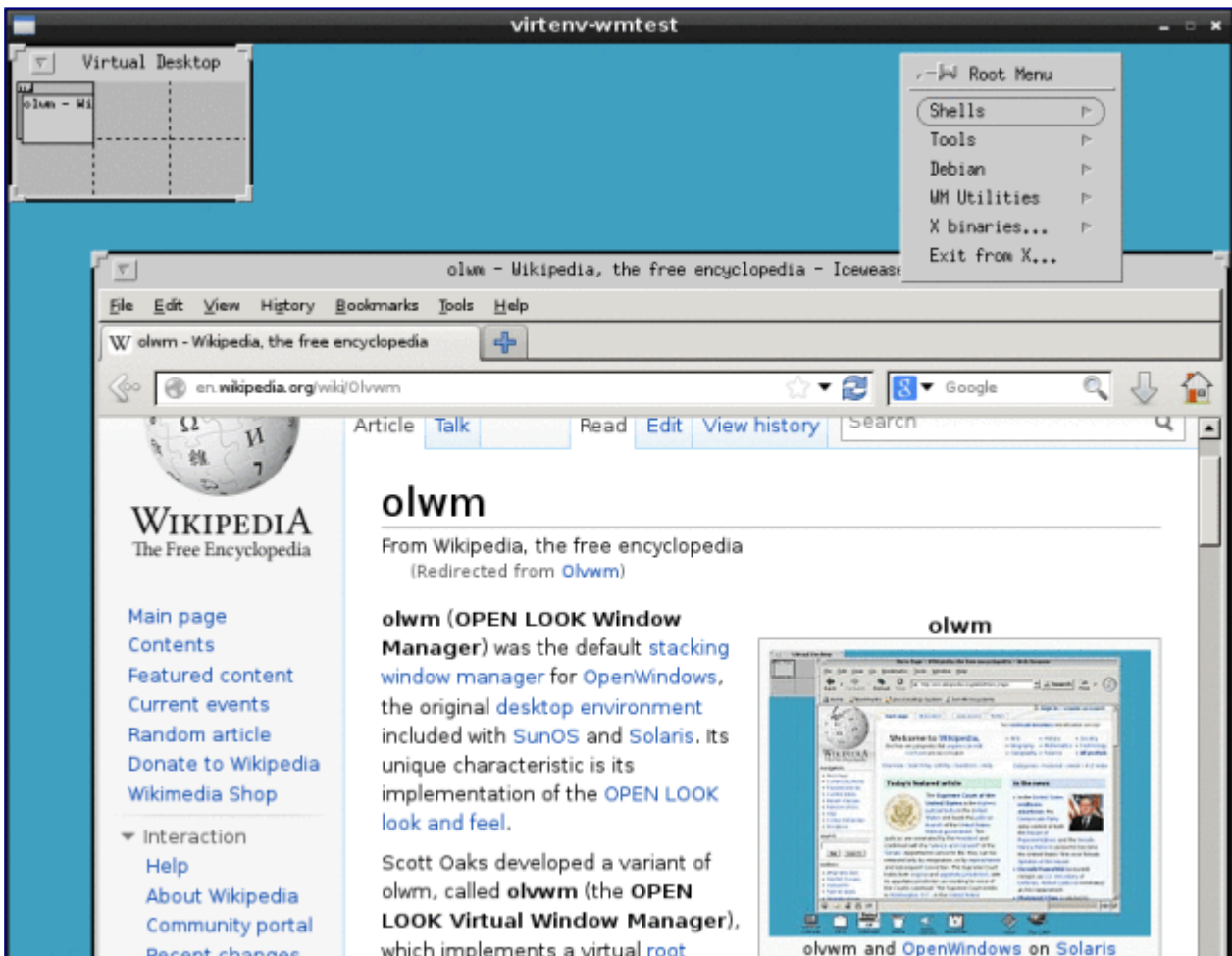


miwm window manager

## olvwm

Around 1.2 MB things get shiny. OPEN LOOK Window Manager was the default window manager in SunOS and Solaris. The open-source version is [olvwm](#) (OPEN LOOK Virtual Window Manager), developed by Scott Oaks. It implements a root window with dimensions greater than those of the video display, and it works on any X11R5 or X11R6 systems.

Install olvwm in Debian/Ubuntu as *apt-get install olvwm*. Start it as *olvwm*. It runs in 1.2MB memory.

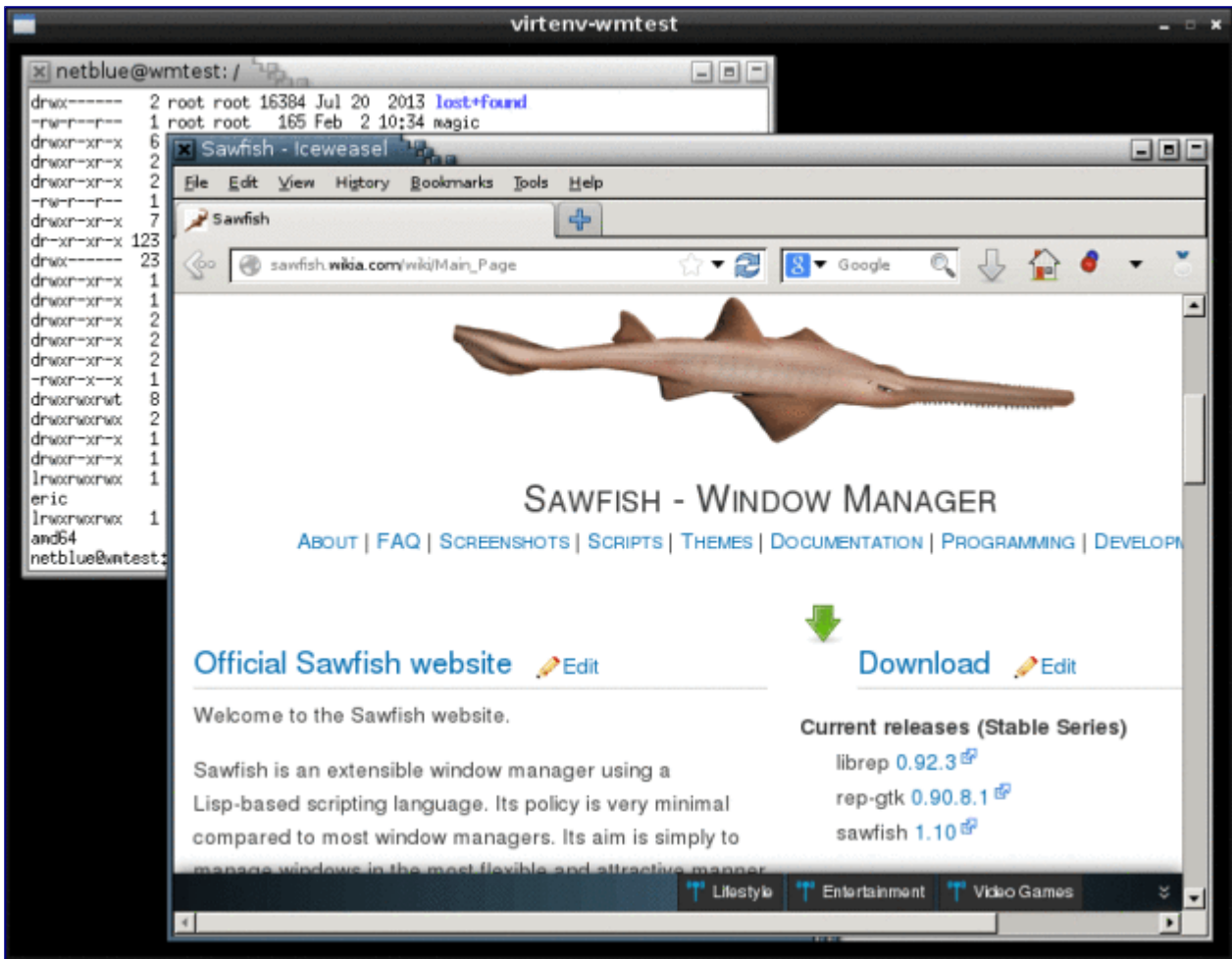


olvwm window manager

## Sawfish

As more and more features are added, the memory increases. This time we look at [Sawfish](#), a themable window manager using a Lisp-based scripting language. Providing only window manager capabilities and nothing else, Sawfish is usually combined with other programs such as [wbar](#) or [LXPanel](#).

On Debian/Ubuntu install sawfish as *apt-get install sawfish* (*yum install sawfish* on Fedora). Start it as *sawfish*. It runs in 3.1MB of memory.

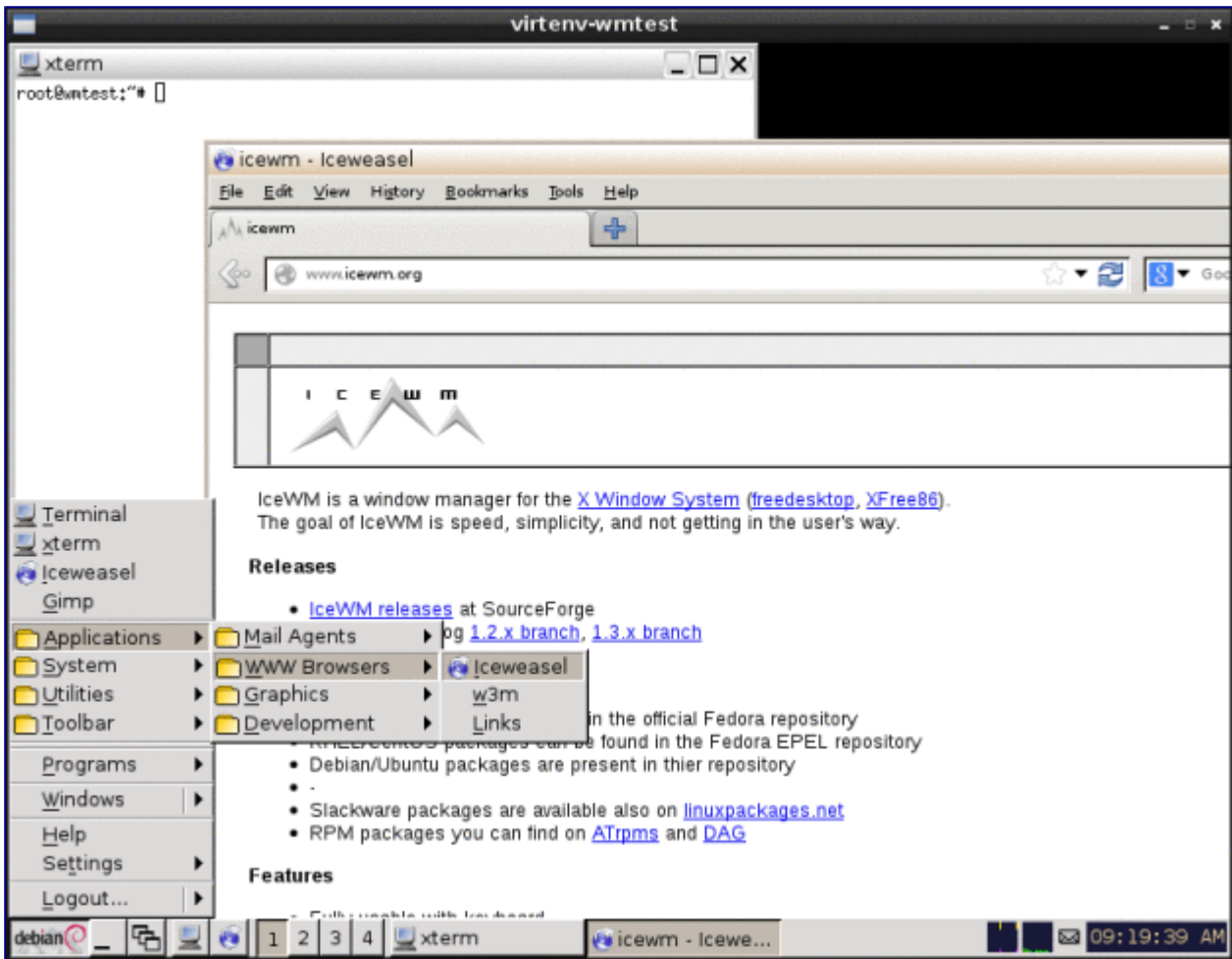


Sawfish window manager

## IceWM

4.5MB of memory later we have a window manager with taskbar, menus, network and CPU meters, mail check and clock. This is [IceWM](#), a fast and simple window manager. It comes with a number of themes that allows it to imitate Windows 95, Windows XP, Windows 7, OS/2, and Motif.

On Debian/Ubuntu install IceWM as `apt-get install icewm` (`yum install icewm` on Fedora). Start it as `icewm`. It runs in 4.5MB of memory.

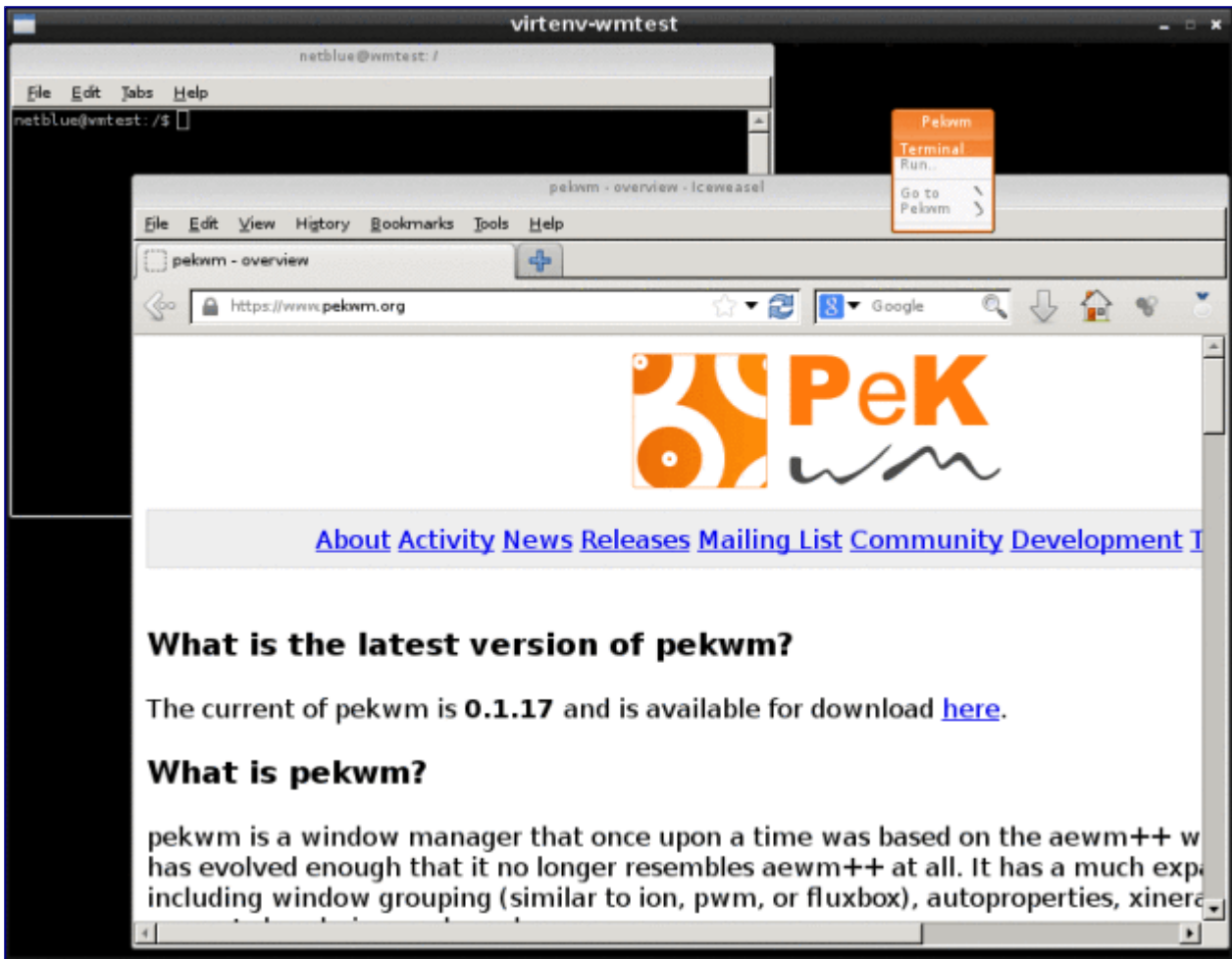


IceWM window manager

## PekWM

[PekWM](#) (The Pek Window Manager) is a window manager based on [aewm++](#). It supports window grouping, auto properties, xinerama, keygrabber, keychains, and much more.

On Debian/Ubuntu install PekWM as `apt-get install pekwm` (`yum install pekwm` on Fedora). Start it as `pekwm`. It runs in 5.3MB of memory.



PekWM window manager

## Mutter

[Mutter](#) (Metacity Clutter) is the current Gnome 3 window manager. Based on [Clutter](#) library, Mutter supports OpenGL and a large number of visual effects. Today, Mutter is in the process of becoming a Wayland compositor.

Mutter is an excellent choice for building Gnome-like desktops. Give it a try if you are into Gnome 3 look and feel, but you don't like the functionality Gnome Shell has to offer.

On Debian/Ubuntu install Mutter as *apt-get install mutter* (*yum install mutter* on Fedora). This will bring in over 50MB of various Gnome-specific packages on your hard disk. Start it as *dbus-launch --exit-with-session mutter*. It runs in 18.6MB – I've tested it on a Nvidia card running Nouveau driver.



Mutter window manager

## KWin

[KWin](#) is the default window manager in KDE desktop environment. It is a compositing window manager, currently available backends include XRender and OpenGL. Like Mutter, KWin is also in the process of becoming a Wayland compositor.

On Debian/Ubuntu install KWin as *apt-get install kde-window-manager* (*yum install kwin-gles* on Fedora). This will bring on your hard drive over 170MB of various KDE packages. Start it as *dbus-launch -exit-with-session kwin*. On my computer it runs in 41MB of memory (Nvidia card, Nouveau driver).



KWin window manager

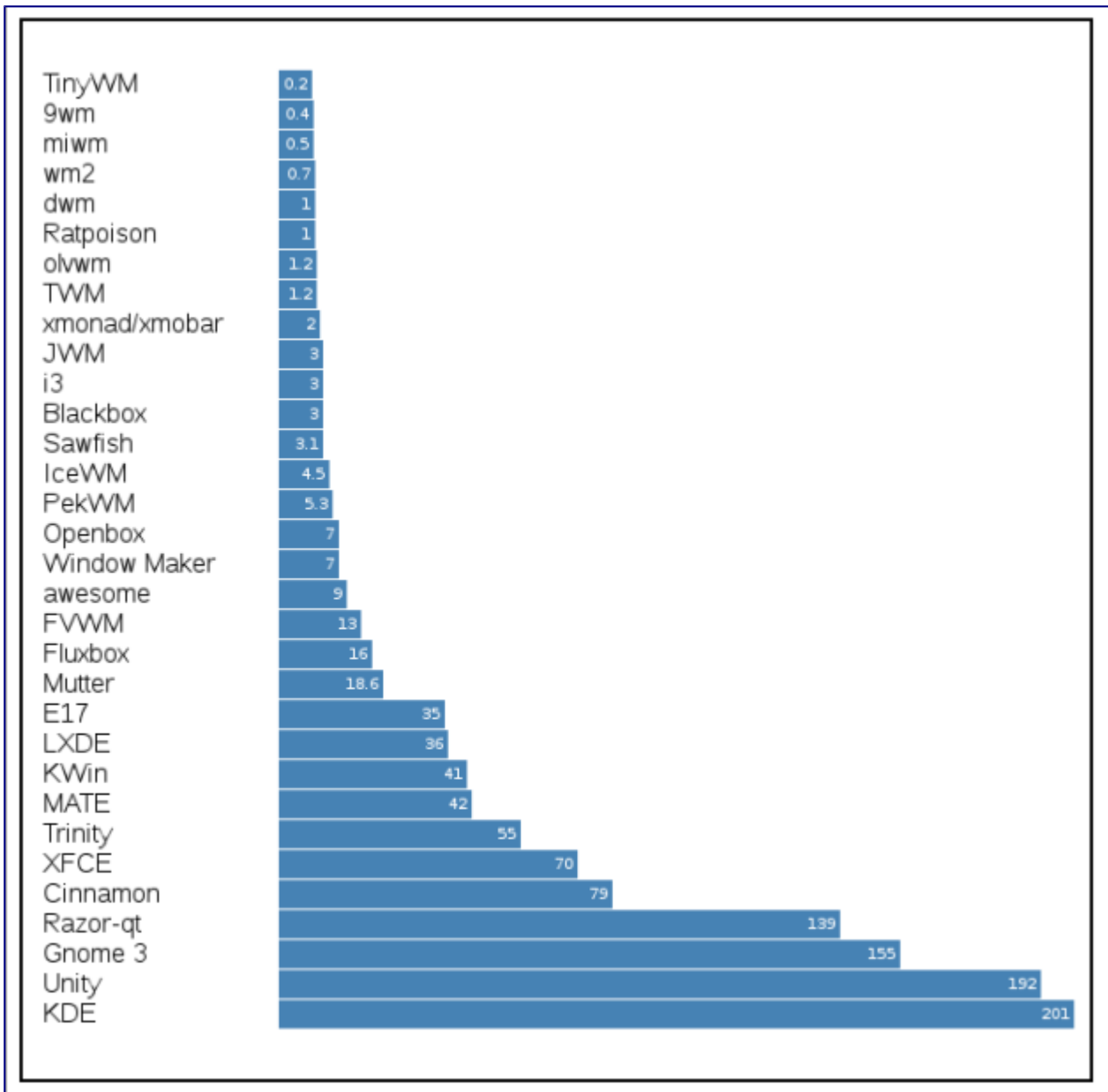
## Conclusion

I have updated the chart with the new numbers. Keep in mind, it reflects software versions as of Ubuntu 12.04 and Debian 7 “wheezy”. Don’t take them too seriously, the chart is more about choice than anything else:

- In 2MB of memory you get tiling window managers suitable for power users and vi experts.
- Around 10MB you have a bunch of themable window managers with taskbars and menus. For many users this is enough, although we are still in power user territory.
- OpenGL-enabled window managers start around 20MB and bring in shadows, transparency and lots of cool visual effects.
- From 40MB up we have the classic lightweight desktop environments. These DEs are becoming more and more popular as computer hardware update cycles lengthen.
- The latest and greatest cutting-edge desktop features can be found starting from 150MB.

Enjoy your Linux desktop!





WM/DE Memory (MB)