

# Jo's alternative desktop ROCK AROUND THE CLOCK

JO MOSKALEWSKI



Sometimes your internal clock is not enough. And anyone who prefers not to wear a watch on their wrist can simply distribute clocks about their environment. The Linux desktop is the perfect place to start.



## The unwritten law

Anyone who has spent long enough over the past few years sitting in front of Microsoft Windows has accepted one tool on its desktop as part of his mental furniture. The clock is on the right at the bottom in the tray of the taskbar. KDE follows this pattern (GNOME, with its more recent versions, no longer does) – it appears to be a current standard. But what do you do if you want to automatically hide this toolbar, but still want to know the time? And nor, by any means, does every UNIX desktop have this type of start menu or taskbar with integrated timepiece.

Nevertheless, GNOME and KDE turn out in this instance to be considerably more flexible than Windows: Their clocks are swappable applets, which can be moved, configured, removed or added. But even these options for the two environments do not exhaust the possibilities on a UNIX desktop: Larger clocks, which are easier to read, can still be placed anywhere on the desktop and thus make more room in any Start menu which may be in use. In any case, anyone who, in the old UNIX tradition, screws his desktop together out of umpteen individual tools, must certainly give some thought to the right clock. But separating the wheat from the chaff can be a very time-consuming venture, though constructing a clock serves for young programmers as a test bed for later successful projects. A small and by no means complete overview of some interesting clocks and their specialities is given here.

## Basic training camp

First of all there are the two simple clocks which belong to the X Window system, *xclock* and *oclock*. Both are thus available practically everywhere and can be configured completely via start parameters or Xresources.

While *oclock* presents itself as a simple, analogue and round clock (although without window frames and with a transparent background!), there is a strange feature in the square *xclock*, called up with the suffix "-digital", this also starts as such. But there is no need to find any deeper meaning behind this. Both display variants are very unfriendly and outmoded in appearance – essentially, they can also only be configured in terms of their foreground and background colours. If you want to use your own colour scheme, you'd be better off with the plain *oclock*. Here each element can be assigned its own colour – whether it's the hour hand, or the numberless dial frame.

### Jo's Desktop

*Only you can decide how your Linux desktop looks. With deskTOPia we regularly take you with us on a journey into the land of window managers and desktop environments, presenting the useful and the colourful, viewers and pretty toys.*



Figure 1: xclock's standard look

Wed Apr 18 21:29:12 2001

Figure 2: xclock -digital

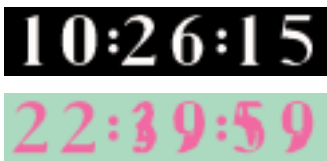


Figure 3: Round object: o'clock



[left]  
Figure 4: Freshly painted: o'clock

[right]  
Figure 5: Stylish: buici-clock



[top]  
Figure 6: xdaliclock

[below]  
Figure 7: Colorcycling & overlapping numbers

Fig. 8: Two-fold copy of the d'clock

## Punctuality

A far more elegant hermaphrodite is the *buici-clock*, which mainly comes with the Debian distribution (the program author is part of this project). Simply round and frameless – like *xclock*, this too can only be configured in terms of position and size. The colours are pre-set and unchangeable – but since its appearance is based on a railway station clock, this is not a failing, but deliberate.

## Finishing touch

Both *buici-* and *oclock* are really easy to use on the desktop, all they need are the clearly-defined area of the clock face. There is no background painted around them by these applications, and nor is there a window frame to spoil the picture. The digital clocks described below are more difficult to design here. Since these are usually letters, these have a text colour as well as a background colour. Certainly instead of the latter it would be nicer to have simple transparency – but each and every programmer seems to be shirking this task more or less successfully. This is where the only thing that will help is a background colour that is similar to the desktop image, or at least matching. Plus, a completely different problem now arises. Since they are not transparent anyway, the creators also forgo an unframed representation. Anyone who now wants to not only click away or minimise his clock, but also wants to look at it on all virtual desktops, would be well advised to take a look in the Window Manager documentation. If this offers no solution, *xnodecor* may help (see box).



### Window frames? No thanks

If you find that your favourite clock is adorned with a nice window frame by your window manager and the latter offers no option for changing this condition then we recommend that you use *xnodecor*. This is very easy to use. To start the clock, you should not use the works-installed Autostart function of many desktop interfaces, but the user's own start file *~/.xinitrc* or *~/.xsession*. This will start the clock first, and thus hide it from the window manager, which is only then to be started. Example of a start file:

```
dclock -geometry -0-0 &
xnodecor -w d'clock
twm
```

## Goliath

With the *xdaliclock* the program author has attempted to get round these problems at the nuts and bolts end – but by doing so, he has also created a few new ones. But the nice thing about this clock is that the numbers fade into each other (also possible when colouring). The only thing is, the numbers have ended up really huge. Unfortunately this is also embedded into the program, so that there is no longer any option here for individual adaptation. But the author must be given credit for having planned in at least one “transparent” mode – and also the windowless representation on the desktop background – from the ground up. Unfortunately, it falls a bit short in the execution, which can easily be checked with a:

```
xdaliclock -root -transparent
```

*xdaliclock* immediately takes over the entire desktop – and at the same time it comes out, not transparent, but coloured. If the clock is only called up with the option *-transparent* and the drama with the windows is left to the listing in the box, this only works for a short time: *xdaliclock* reports back in lively fashion every minute to the window manager, and the magic of the transparency is countered by marked representation errors.

## David

For sheer undiluted joy, however, go to *dclock*: This, too, is digital, overlaps (apart from the seconds) the figures with each other, is freely configurable in terms of size, scope of representation and colour – and also offers an alarm function. One would almost be tempted to say its perfect, but unfortunately there is no transparent mode. Figure 8 shows this clock once with the standard defaults in the window, and once completely configured and integrated into the desktop (including implementation of *xnodecor*). The following command parameters are especially interesting with this clock:

- *-bg [color]*: background colour
- *-fg [color]*: foreground colour
- *-led\_off [color]*: colour of the inactive segments of the LED display
- *-fn [font]*: font for the date
- *-geometry -0-0*: off in the corner with ...
- *-miltime -date "%A, %d %B %Y%":*
- *-seconds*: displays seconds
- *-noblink*: blinking ":" avoid
- *-notails*: 6 & 9 without "crossbars"
- *-fade -fadeRate 100*: fade in the figures
- *-slope [X]*: place figures slanted by X%
- *-smallsize 0.5*: show seconds half as high as minutes
- *-nobell*: switch off half-hourly "alarm" ■