

How to use your computer as an alarm clock

Note: This article is now completely outdated: do not use it for anything serious.

1 Introduction

For a long time, I had been using that crappy FM radio clock which can only wake you up by connecting to some of the lame stations there are in the deserted area you live in. Why go through such pain while you have your computer two meters away, with your 10 000 songs collection and a connection to your favorite web radio ? I decided it was enough. The next steps I took are detailed in this mini-howto.

Important: This howto only details the procedure if you actually want your computer to boot and wake you up. There are far more convenient solutions if you use hibernation, or if your computer is on all the time (I personally use cron jobs, but there are also dedicated programs like KAlarm¹). Actually, those solutions may even apply to the case this howto covers but I haven't managed to get anything else working. If you have, please let me know².

2 Pre-requisites

I detail the procedure for a UNIX-like system with KDE 3.5 installed. If you have another system, you will have to do some googling to find the corresponding features.

You will also need a music player easily controllable from the command line (e.g AmaroK does not do the trick, as it can only be controlled via DCOP, and the DCOP server is not started until after you login). I use mplayer. One of my script is written in Perl³, so you will need it as well. Also, you will need some X-window interface generator, such as Xdialog (the one I use). It will be used to generate an interface you will use to stop the music once it has started.

Finally, do not expect all this to work for you on the first place, as every installation is different. You will probably have to change some small things here and there in my scripts.

3 Overview

You will need your computer to automatically boot at the time you want to wake up. To do that, the first step is to go into your BIOS and configure it to power on your PC at some given time. If you have more specific needs, you can also try wake-on-LAN⁴. I have never done it myself so I cannot give you any advice.

¹<http://www.astrojar.org.uk/kalarm/>

²http://svasey.org/about_en.html#contact-information

³<http://www.perl.com>

⁴<http://en.wikipedia.org/wiki/Wake-on-LAN>

Now the problem is as follows: you need the music to play immediately after you PC starts. You do not want to login to do that. One simple solution is the auto-login feature provided by KDE and GNOME. You will simply be logged in without having to supply a password, and can then put my scripts in the Autostart directory and it will be that simple. It is not secure though, because anyone and not only you can login using that feature.

Another solution is to add a new RC script. This way the music will start but you will not be able to stop it until after you log in...

So starting everything before or after the login does not seem very convenient. There is only one possibility left: start it at the same time as the login manager. If you use a graphical login manager, you will then be able to stop the music by just clicking on a popup and then proceed to login normally.

4 Launching the script with the login manager

Warning: changing the login manager configuration carelessly can compromise the security of your system. Be sure to read the fine manual before attempting any change and never, never take my words for granted ;-)

I assume you are using kdm as a login manager. It keeps its configuration files `/usr/kde/3.5/share/config/kdm` on my system. The file which interests us is `Xsetup`. It is a script run just before the login dialog appears. Here is what I have in mine:

```
#!/bin/sh
# Xsetup - run as root before the login dialog appears

/usr/local/sbin/startup.pl

exit $?
```

`/usr/local/sbin/startup.pl` is to be replaced by the path where you will install my `startup.pl` script. As it is going to be run as root, you'd better **put it in a place where it cannot be modified by ordinary users**.

5 Installing and customizing the script

Here is the tarball containing the two scripts you will need⁵.

5.1 Customizing startup.pl

The `startup.pl` script is the one you will have to customize most often. It checks if the current time corresponds to the time it should wake you up, and launch the music (calling the `startup-mplayer.sh` script). Afterward, it generates a popup using `Xdialog` and wait for you to close that popup or for it to timeout. Once that is done, the music will be stopped.

`startup.pl` uses the `Proc::Simple` module. Just copy the `Proc` directory into your perl library path.

⁵http://svasey.org/download.svasey.org/wakeup/wakeup-1_0.tar.gz

Then open `startup.pl` and edit the `XDIALOG` and `MPLAYERSTART` according to your system. You can then edit the `PLAYLISTS` variable. The first playlist appearing will be played first. You can also change the `TIME` variable if you are not satisfied with the default.

Now comes the interesting part. Since you will not want the music to be played every time you boot your computer, but only at some specific time and days, you should take care of the following variables: `REGTIME`, `EXCLUDETIME` and `OTHERTIME`.

Each of those three objects is described by the comments above it. For each object, there are five keys (day, month, min, hour, wday) you should take care of. Their value is a two-elements array, describing an interval between which the music will (or won't, depending on the object) be played. I believe I explained enough about that file. Go ahead, do your changes and **test them**. If you have any more questions look at the code or send me a mail⁶.

5.2 Customizing `startup-mplayer.sh`

It should not be too difficult to understand that script. Change the `MPLAYER` variable to the path to your music player executable. If you change it, you should also change the command line options it is given.

The script also temporarily changes the volume. I assume you are using `alsa`. If you are not, you will have to change that part as well. In any case, be sure to adapt the volume to your needs.

The only things `startup.pl` expects concerning that script is that it takes some path to playlists as arguments, and correctly traps `SIGINT` and `SIGTERM` (i.e `startup-mplayer.sh` will stop the media player if it is killed)

6 Last words

Now that you have set everything up, there is only one thing to do: *test, test and test*. Do not forget that **this will be run as root** so you'd better be careful.

If you managed to get it working, I'd like to hear from you⁷.

⁶http://svasey.org/about_en.html#contact-information

⁷http://svasey.org/about_en.html#contact-information