

Software TO THE MINUTE

CHRISTIAN PERLE



There are thousands of tools and utilities for Linux. We take the pick of the bunch and suggest a little program each month that we feel is either absolutely indispensable or unduly ignored. This month we put the spotlight on gatO.

With the utilities **cron** and **at**, every Linux system has the use of control mechanisms which are accurate to the minute. So you can have yourself trumpeted awake every morning at seven by soft MP3 music (provided the fan in the computer is quiet enough) or postpone the tedious calculation of a **Raytracing** scene to the middle of the night.

Quarter to...

Before you can enjoy using the graphically usable *at*, a few more conditions have to be met. So you will need:

- the **GTK** packet (*libgtk* etc.), plus
- the GTK Development packet (*gtk-devel* etc.) together with
- **autoconf** and **automake**

cron: Serves for regular timed executions of any console programs (Cron-Jobs) and thus takes care, among other things, of updating the databank for the *locate* command.

at: Unlike *cron*, *at* performs its duties (At-Jobs) one time only at a specific time. Any outputs from the program started are sent as mail to the owner of the At-Job. But using the *at* command is rather difficult for beginners, because you have to comply with a specific format when entering details of the date. A graphical front-end would be nice here – which is precisely what *gatO*, from Antonio Luis Delgado Gonzalez, is.

Raytracing: Procedure for photo-realistic calculation of three-dimensional images, which is highly processor intensive.

After a visit to the gatO homepage (<http://www.arquired.es/users/aldegado/proy/gato/default.htm>) you will be better off by one file (*gato-0.6.6.tgz*). Unpacking and installation of the program functions as follows:

```
tar xzf gato-0.6.6.tgz
cd gato-0.6.6
su (enter root password)
make
cd ..
rm -rf gato-0.6.6
exit
```

Twelve!

If everything has worked, the executable file *gato* will now be in the directory */usr/local/bin*. Since this is usually included in the **PATH** variables, it is sufficient to enter *gato &* in an *xterm*, *K* or *GNOME* terminal to start the program. The main window that now appears is shown in Figure 1.

It is split into two main areas. The left half shows the tasks already assigned, while on the right new At-Jobs are entered. In Figure 1 there are two tasks in the queue, which are to be executed on 8.2.2001 at 23:30 and on 9.2.2001 at 7:00.

Job agency

We want to add another job to these. Since an At-Job rarely consists of the call-up of a single program, it is best to create a text file containing the desired commands. For example save the following lines in a text file under the name *findexec* in your home directory:

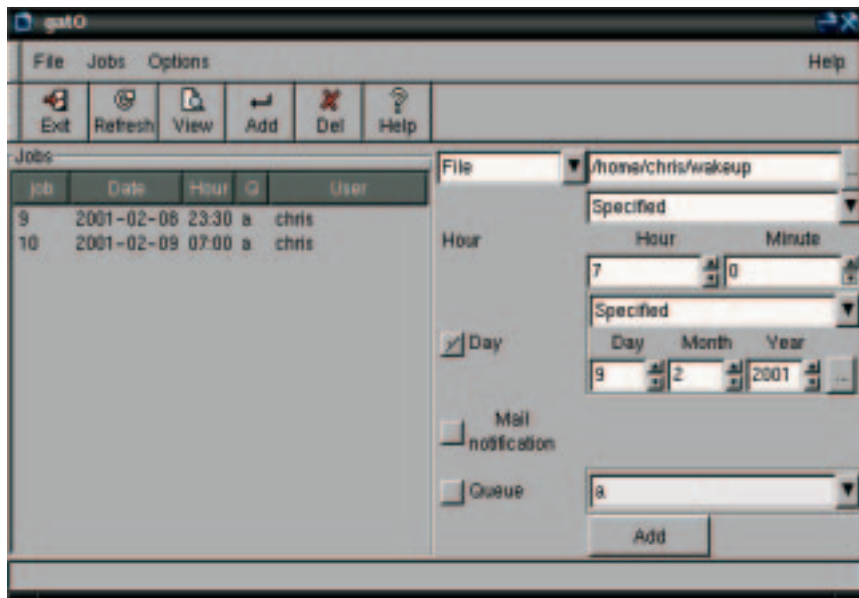
```
cd $HOME
find . -type f -perm +111 > exec.log
echo "Executable files listed in exec.log."
```

This three-liner searches, starting from the home directory, all executable files and stores the result in the file *exec.log*. The file *findexec* has no need of execution rights, since gatO transfers its content to *at*.

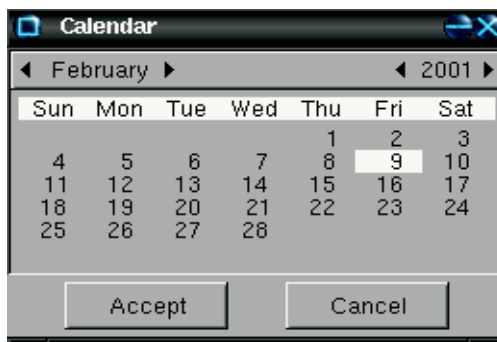
In the right half of the gatO main window, change from *Command* to *File* and click on the "... " button in the same line. A selection box now opens for the file whose content is later to be executed by *at*. Alternatively, you can also enter the file name direct in the box on the right next to *File*.

Set the time when this job is to be done with the boxed situated underneath. If the checkbox next to *Day* remains unticked, the time specification relates to the same day so that only hours and minutes play a role. If *Day* is selected, on the other hand, you can also use the "... " button here to set the date easily with a mini-calendar (Figure 2).

If you do not want to enter the time or the day in figures, you can use the two *Specified* buttons to enter details such as *midnight*, *teatime* or *tomorrow*, *Mon*, *Tue*. In this case the figures underneath are ignored. If a job does not have an automatic output but you are still interested in a report by mail, select the *Mail notification* button. Finally, click at the bottom right on *Add*, to include the job in the queue.



[top]
Figure 1: gatO main window



[left]
Figure 2: Calendar

Forgetful?

If you have already entered several jobs, but no longer recall what the individual assignments do, you just have to mark the appropriate entry on the left side and click on the *View* button. A separate window then displays the commands of which the job consists, with the first part consisting of internal *gatO* lines and the actual job only coming after that (Figure 3).

Jobs can also be removed from the queue using the *Del* button – at least as long as they have not already been done. The *Refresh* button serves to redisplay the job list. When this happens the jobs

which are no longer in the queue since the last *Refresh* disappear, because they have either been deleted or executed. The *Exit* button self-explanatory.

If you can handle gatO well, there is more good news for all those who found no suitable program for managing Cron jobs in the overview article last month: Antonio Luis Delgado Gonzalez is also writing a program in this category – *gcrontab*, which can be found at <http://www.arquired.es/users/aldegado/proy/gcrontab/>. ■

GTK: The "**Gimp ToolKit**" is a programming library that facilitates the creation of menus, dialog boxes and other window elements under X11. It was originally written for the graphics program *Gimp*, but has now become the basis of many other programs.

autoconf, automake: Tools for the automatic creation of a **configure script**.

configure script: Serves for the automatic evaluation of properties of the system on which a source text is to be compiled. So *configure* finds out automatically, which C Compiler is installed. With the aid of this tool, it is possible to compile cross-platform software on various (mostly Unix) systems, without having to make any changes by hand.

PATH: This variable lists directories separated by a colon. These are searched by the shell for commands that the user enters without path specification. So for example *ls* is found as */bin/ls*.

Figure 3: Job Ad

