at - Task Scheduling

at is a task scheduling tool which one-time enables tasks to be performed on your system. The **batch** is like **at** but permforms the one-time tasks when system load permits it, all command and functions of **at** can be replicated by **batch** with the same arguments.

Related Commands

The following commands are included to monitor the **at** command:

- **at** - The program used to submit a one-time task.
- **atq** - The program used to review the queue of jobs submitted with **at**

.

- **atrm** - The program used to remove jobs from the at queue.

at Command Use

The **at** command can be issued to allow you to write a custom script or point it to a pre-built script. In addition, the broad time specifications allowed increase the complexity.

The *-m* argument can be added to any command telling the daemon to mail the user on the local system advising of the completion of the task.

Scheduling an Existing Script

This example will tell the **at** application to run the script */home/user/testscript.sh* at 4PM today and e-mail the user upon completion.

```
user@darkstar$ at 16:00 -m -f /home/user/testscript.sh
```

As you can see the *-f* argument means **file** as it points to the file that will be executed.

Scheduling a New Action

As seen below, when no file is specified the **at** application opens a prompt for the user to enter the new script.

```
user@darkstar$ at 19:40 -m
warning: commands will be executed using /bin/sh
at> ping -c 4 www.google.com
at> echo $?
```

```
at> <EOT>
job 4 at Sun Dec 30 19:40:00 2012
```

The user will need to type the sequence of commands that will be issued within the prompt, upon completion you close the prompt with the Ctrl+d

The command above instructed **at** to ping google 4 times at 7:40PM, return the application status and e-mail the results to the user.

Be sure to note the job number, it is the only visible identifier for the job and must be referenced if you wish to remove the job later.

Reviewing Queued Tasks

The command **atq** lists the user's pending jobs. The output is Job Number, date, hour, queue and username.

```
user@darkstar$ atq
4 Sun Dec 30 19:40:00 2012 a user
```

Once the task number has been noted you can review the commands that are included in the job by issuing the command *at -c {jobnumber}*, and example is shown below.

```
user@darkstar$ at -c 4
#!/bin/sh
# atrun uid=1000 gid=1000
# mail user 1
umask 22
....Environment Details....
cd /home/user || {
   echo 'Execution directory inaccessible' >&2
   exit 1
}
ping -c 4 www.google.com
echo $?
```

Removing Queued Tasks

The command **atrm** removes jobs from the pending queue, it does not present any confirmation that the job is removed so it may be advisable to run **atq** after **atrm** to confirm the job has been removed.

```
user@darkstar$ atrm 4
```

Further Resources

- at manual page
- BrunoLinux

- http://sathyaphoenix.wordpress.com/2009/01/18/using-at-command-to-schedule-jobs-in-linux/

Sources

* Originally written by mfillpot

howtos, task scheduling, author mfillpot

From:
https://docs.slackware.com/ - **SlackDocs**

Permanent link:
https://docs.slackware.com/howtos:software:at

Last update: **2015/04/14 01:47 (UTC)**