BASH

SECTION 10

SCHEDULING AND AUTOMATION

SECTION CHEAT SHEET

THE "AT" COMMAND:

The at command is the most basic way to schedule tasks in Linux. It allows you to run a script at a *one-off point in time*.
You **cannot set up repeated scheduling** with the at command.

Starting the at service:

Command to install the at command on Ubuntu:

```
sudo apt install at
```

To check if the at service running:

```
service atd status
```

To start the at service

```
service atd start
```

To stop the at service

```
service atd stop
```

To restart the at service

```
service atd restart
```

SCHEDULING JOBS WITH “AT”

Syntax to schedule one or more commands:

```
at <time> <date>
```

This will cause the at command to open a prompt where you can add as many commands as you like to be run as part of this job.

You end the prompt with ctrl + d once you have added all your commands.

Syntax to schedule a script:

```
at <time> <date> -f /path/to/script
```

Some Useful Options:

-l	List all jobs that we have scheduled
-r <id>	Remove the job with id “id” from the schedule

Expressing dates and times

Some Sample Time Formats:

HH:MM	Run at HH:MM in 24-hour format
10pm	Run at the upcoming 10 pm timeslot (either today if it's currently before 10pm, or tomorrow if not.)

Some Sample Date Formats:

7/12/2021	Date in month/day/year. This represents 12th July 2021
7.12.2021	Date in day/month/year format.
next week	Exactly 7 days from now
+ 3 days	3 days from now

Hint: Please view the at command's man page by running "man at" for more ways to express timepoints.

Limitations of the "at" command:

- The at command will only execute jobs at the scheduled time if your PC is turned on and the at daemon is running at that time.
- There is no way to set up recurring jobs. Jobs scheduled by the at command only run once.

THE "CRON" COMMAND:

Like the at command, cron requires a daemon service, crond, to be running on your system in order to work.

To check if cron is running:

```
service crond status
```

To start cron:

```
service crond start
```

To stop cron:

```
service crond stop
```

To restart cron:

```
service crond restart
```

EDITING CRONTABS

Crontabs are files that express, in table format, the jobs that we want to schedule and at what time.

Each user has a crontab that contains the jobs that they want to run from their user account.

To edit the current user's crontab:

```
crontab -e
```

Important: Always use crontab -e to edit your user's crontab. This will ensure that the cron service is restarted and that your changes take effect.

Crontab expression syntax

Each row in a crontab expresses a new job, and each row is made up of 6 columns separated by white space.

Each element of the expression must be separated, but the amount of space between each column doesnt matter and does not have to be consistent

Crontab Column Title						
	minutes	hours	days of the month	month	days of the week	command
Valid Values	0-59	0-23	1-31	1-12	0-6 or MON-SUN	Your command or PATH to your script

Important: You must ensure that the scripts referred to in the “command” section have been given execution permissions!

Some handy shortcuts

Character	Example	Meaning
*	n/a	Putting * in the hour column is the same as entering the numbers 0-23 Putting * in the minute column is the same as entering the numbers 0-59 Putting a * in the days of the week is the same as entering the numbers 0-6
,	1,5,8	Enter the values 1, 5, and 8 into the current column
-	1-8 MON-WED	Enter the values 1,2,3,4,5,6,7,8 into the current column. Enters the values MON,TUE,WED into the current column.

Hint: You can check whether your crontab expressions match your intentions by using a great tool called crontab.guru

Hint: Run the command `man crontab` for more information on crontabs

CRON DIRECTORIES:

Cron directories are folders on your system where you can place a group of scripts that you want to run at the same frequency.

Preconfigured directories:

/etc/cron.hourly	Scripts in this folder will run once per hour
/etc/cron.daily	Scripts in this folder will run once per day
/etc/cron.weekly	Scripts in this folder will run once per week
/etc/cron.monthly	Scripts in this folder will run once per month

Important Note 1: Any script placed within these folders will be run by the root user

Important Note 2: Scripts placed within these folders must not contain "." characters in the filenames (e.g. .sh)

Important Note 3: Like all cron scripts, you must ensure that scripts you place in these folders have execution permissions! Otherwise, they won't run!

Creating your own cron directories

Any folder can become a cron directory. All you need to do is to add a new row in your user's crontab with the time you want the directory's contents to run, and then enter the following in the command column:

```
run-parts --report /path/to/directory
```

Limitations of Cron:

Cron will only execute jobs if your PC is turned on and the cron daemon is running at the scheduled time.

ANACRON:

The advantage of anacron is that it has the ability to monitor if a cron job has been run or not, and if not, then run it when this lapse is discovered.

By default, there is only one anacrontab on a system by default. Anacron's crontab is located at /etc/anacrontab.

As /etc/anacrontab requires is a system-wide configuration file, root privileges are required to modify it.

Therefore, to modify the anacrontab, you must run:

```
sudo nano /etc/anacrontab
```

Anacrontab syntax

Each row in an anacrontab is made up of 4 columns as follow:

Anacrontab Column Title				
	period	delay	job-identifier	command
Meaning	How many days between each time your command is run. E.g putting a "1" here would make the command run every day	The delay in minutes from when anacron starts to when this command is run	The name given to the command in the anacron logs. The job identifier can contain any characters except blanks and slashes.	The command you want to run, or path to your script.

Hint: Please run the command man anacrontab for more information on anacrontabs

Limitations of Anacron

- Unlike cron, you cannot run anacron jobs any more frequently than daily.
- Anacron will only recover 1 missed job!