



# **Part 39**

-

# **Adding crontab entries**

## Steps to create cron job

### Give crontab privilege

Before we start we need to give crontab privilege to the respective user. For the sake of this article I will create some sample cron job for root and pi user, so I will assign permission for these two users. Append the username to `/etc/cron.allow`

```
# cat /etc/cron.allow
root
pi
```

### Create cron file

Create a cron file for root user

```
# touch /var/spool/cron/root
# /usr/bin/crontab /var/spool/cron/root
```

Create a cron file for pi user

```
# touch /var/spool/cron/pi
# /usr/bin/crontab /var/spool/cron/pi
```

### Cron line explanation

```
* * * * * "command to be executed"
- - - - -
| | | | |
| | | | ----- Day of week (0 - 7) (Sunday=0 or 7)
| | | ----- Month (1 - 12)
| | ----- Day of month (1 - 31)
| ----- Hour (0 - 23)
----- Minute (0 - 59)
```

### Schedule your job

I will create some dummy jobs. To give a demonstration I will schedule a job to clear temporary files every midnight for both the user

```
# echo "0 0 * * * rm -f /tmp/root/*" >> /var/spool/cron/root
# echo "0 0 * * * rm -f /tmp/pi/*" >> /var/spool/cron/pi
```

### Validate the cron job content

Here you can use `-u` to define the username for which you wish to perform the cron action/

```
# crontab -u pi -l
0 0 * * * rm -f /tmp/pi/*

# crontab -u root -l
0 0 * * * rm -f /tmp/root/*
```

So our cron jobs are updated successfully for both root and pi user.

**Note:** You do not need to restart your crond service for the new changes.

```
# service cron restart
```

## Script to create cron job using bash shell script

Let us put them all together in a script

```
#!/bin/bash
if [ `id -u` -ne 0 ]; then
    echo "This script can be executed only as root, Exiting.."
    exit 1
fi

case "$1" in
    install|update)

        CRON_FILE="/var/spool/cron/root"

        if [ ! -f $CRON_FILE ]; then
            echo "cron file for root doesnt exist, creating.."
            touch $CRON_FILE
            /usr/bin/crontab $CRON_FILE
        fi

        # Method 1
        grep -qi "cleanup_script" $CRON_FILE
        if [ $? != 0 ]; then
            echo "Updating cron job for cleaning temporary files"
            /bin/echo "0 0 * * * rm -f /home/pi/cleanup_script.sh" >> $CRON_FILE
        fi

        # Method 2
        grep -qi "cleanup_script" $CRON_FILE
        if [ $? != 0 ]; then
            echo "Updating cron job for cleaning temporary files"
            crontab -u pi -l >/tmp/crontab
            /bin/echo "0 0 * * * rm -f /home/pi/cleanup_script.sh" >> /tmp/crontab
            crontab -u pi /tmp/crontab
        fi

        ;;

    *)

        echo "Usage: $0 {install|update}"
        exit 1

    ;;
)

esac
```

Here I have shared two methods to update cron job using a shell script for root and pi user. You can validate the changes after executing your script

```
# /tmp/script.sh install
```

Updating cron job for cleaning temporary files

### List the cron jobs

```
# crontab -u root -l
0 0 * * * rm -f /home/pi/cleanup_script.sh

# crontab -u pi -l
0 0 * * * rm -f /home/pi/cleanup_script.sh
```

So all is working as expected.

## **Adding crontab lines via shell scripting**

### ***Adding only***

You can add to the crontab as follows:

```
croncmd="echo hello"
cronjob="0 */15 * * * $croncmd"
(crontab -l && echo "$cronjob") | crontab -
```

### ***Adding & removing***

The method below is immune to multiple insertions, and it lets you change the schedule of an existing entry. Say you have these:

To add to the crontab, with no duplication:

```
( crontab -l | grep -v -i -F "$croncmd" ; echo "$cronjob" ) | crontab -
```

To remove it from the crontab whatever its current schedule:

```
( crontab -l | grep -v -i -F "$croncmd" ) | crontab -
```

Notes:

grep -F matches the string literally, as we do not want to interpret it as a regular expression. We also ignore the time scheduling and only look for the command. So all crontab entries are listed except the ones matching the croncmd. The schedule can be changed without the risk of adding a new line to the crontab.