How to enable pc speaker on UBUNTU 20.04 to make your server audible without additional hardware

Maybe you run a physical Linux server somewhere under a desk or somewhere in your small office / home office which still contains spinning platters (hard disks) or any other hardware that “may fail” at some point. How do you get noticed that your server is in pain ? Send an automatic email ? Good Idea!! Way to go!

Who is reading it and when ?

Independent of that i prefer my physical servers to be AUDIBLE, to be able to **cry for help**, using the almost forgotten “on board pc speaker” that most (but not all) computers still have. But for whatever reason the pc speaker is ofen no longer “active” by default under Linux. So we need to enable it and make sure, that its driver is loaded on every reboot automatically.

This manual shows a way on how i enabled it and how i use it to alert me “audibly” when something critical on the server happens.

1. Install the “beep” tool:

   ```
   apt install beep
   ```

2. Load/Add Linux Kernel Module (driver) for the PC speaker:

   ```
   modprobe pcspkr
   ```

3. Test beep :

   ```
   beep
   ```

4. Nice. Now we make sure that this Kernel Module is loaded on every reboot automatically. Therefor we need to take if from the “Kernel Module Blacklist”:

   ```
   # always make a backup first.
   # make sure it does not end on .conf
   #
   cp -v /etc/modprobe.d/blacklist.conf{,.$(date +%F)}

   # comment out corresponding kernel module
   # from blacklist
   #
   sed -i 's/blacklist pcspkr/#blacklist pcspkr/'
   /etc/modprobe.d/blacklist.conf
   ```

5. reboot and test again if beep still works, without “modprobing” manually again.
6. Now you can do lots of things. Add it to your shell scripts or cronjobs. Whatever you prefer. Here is a little demo on how i watch the RAID health status (alias md raid, mdadm) and use the PC speaker to alert me while daytime by MORSE CODE. :

7. Prepare the shell script which tests the RAID health and does the morse code in case:

/usr/local/sbin/check-raid-status.sh

```
#!/bin/bash

# CHANGE LOG:
#
# 2021-08-01  A.Werner    ADD: wall + console output + new string
#                  clean,checking
#
# 2021-09-05  A.Werner    ADD: new OK string added
#

function dit {
    beep -f 2750 -l 75 -d 50
}

function dah {
    beep -f 2750 -l 175 -d 50
}

function spc {
       sleep .1
}

function s {
    dit
    dit
    dit
        spc
}

function o {
       dah
       dah
       dah
       spc
}

function morse_sos {
    s
    o
    s
    sleep .5
}

mdState=$( /usr/sbin/mdadm --detail /dev/md0 | grep "State :" |
cut -d: -f2 | tr -d ' ' )
```

```
case "$mdState" in
    active|active,checking|clean|clean,checking)
        : # nop
        ;;
    *)
        morse_sos
        echo "$0 WARNING: mdadm reports md0 status: '$mdState' on
$(date)" >&2
        echo "$0 WARNING: mdadm reports md0 status: '$mdState' on
$(date)" >/dev/console
        wall "$0 WARNING: mdadm reports md0 status: '$mdState' on
$(date)"
        ;;
esac
```

8. Make shellscript executable:

```
chmod +x /usr/local/sbin/check-raid-status.sh
```

9. Activate a cronjob that runs the check script every minute while daytime:

```
cat <<'EOF' > /etc/cron.d/raid-monitor-md0
#
# Regular cron jobs to audibly alert admin if
# md (mdadm) raid changes state from "clean"
# using morse code
#
#
#m  h    dom    mon    dow    user    command
5   06-22    *    *    *    root    /usr/local/sbin/check-raid-
status.sh

EOF
```

10. If this does not work for you or is too sensitive it should be no problem to pimp it up relatively easy.

---

linux, cli, ubuntu, pc, speaker, pcspeaker, apt, script, audible, audio, beep

From:
https://awerner.myhome-server.de/ - **Axel Werner's OPEN SOURCE Knowledge Base**

Permanent link:
https://awerner.myhome-server.de/doku.php?id=it-artikel:linux:how-to-enable-pc-speaker-on-ubuntu-20.04-to-make-your-server-audible-without-additional-hardware

Last update: **2022-10-30 16:43**