

Backups and misc cron jobs

Mount USB stick (backup stick)

```
#format SD card  
mkfs.ext4 /dev/sda
```

```
vim /etc/fstab
```

configure fstab to skip mount errors because if USB stick is not available boot fails or is really slow due to mount timeout (default seemsto be 90 seconds). More info at <https://www.bloggerle.de/raspberry-pi-emergency-mode-and-root-account-locked-mounting-usb-disk-with-fstab/>

```
/dev/sda          //mnt/external_usb32g    ext4    defaults,nofail,x-systemd.device-  
timeout=5        0          0
```

```
mkdir -p /mnt/external_usb32g  
mount -a  
  
cd /  
ln -sf /mnt/external_usb32g backup
```

Troubleshooting USB stick file system

```
vim /opt/repairUSBstick.sh
```

```
#!/bin/bash  
umount /mnt/external_usb32g  
fsck -y /dev/sda  
mount -a
```

```
chmod +x repairUSBstick.sh
```

Daily repair cronjob

```
vim /etc/cron.d/repairusbstick
```

```
SHELL=/bin/bash
PATH=/usr/lib/sysstat:/usr/sbin:/usr/sbin:/usr/bin:/sbin:/bin
#daily repairing of USB stick
0 6 * * * root /opt/repairUSBStick.sh > /dev/null 2>&1
```

Daily Backup

Backup runs daily at 18:15 (after daily regular closing of the museum)

/etc/cron.d/backuphost

```
SHELL=/bin/bash
PATH=/usr/lib/sysstat:/usr/sbin:/usr/sbin:/usr/bin:/sbin:/bin
15 18 * * * root /opt/backuphost.sh > /dev/null 2>&1
```

/opt/backuphost.sh

```
#!/bin/bash

CUR="$( cd "$( dirname "${BASH_SOURCE[0]}" )" >/dev/null && pwd )"
DTS=$(date +"%Y%m%d_%H%M")
BUP="/backup"

#rsync switches
# D - keep device files
# o - copy owner rights
# g - copy group rights
# r - recursive
# R - copy with folder structure
# L - copy files behind symlinks
# t - keep timestamp
# l - copy symlinks
# p - keep file permissions
# a = Dgloprt

#installed packages
```

```

dpkg -l > "$BUP"/dpkg.list

#some extra options for all rsync - can be used to debug
#RP="-v --progress -stats"
RP=
RSYNC="rsync $RP"

#home, skel, postfix, apache, motd, ssl, init.d, systemd, resolv.conf, interfaces, ...
$RSYNC -lrR /home/      "$BUP"
$RSYNC -aRL /etc/       "$BUP"
$RSYNC -rptR /boot/     "$BUP"

#cron
$RSYNC -laRL /var/spool/cron/crontabs/ "$BUP"

#user
$RSYNC -lrptR /usr/local/      "$BUP"

#get host-specific backup jobs
$RSYNC -lrptR /var/lib/Repetier-Server/ "$BUP"
$RSYNC -lrptR /usr/local/Repetier-Server/ "$BUP"
$RSYNC -lrptR /usr/local/Repetier-Setup/ "$BUP"
$RSYNC -lrptR /usr/local/share/mjpg-streamer/ "$BUP"

#opt
$RSYNC -lrptR /opt/ "$BUP"

#sh usage of /backup
SIZE=$(cd "$BUP";echo $(du --apparent-size -sh)|cut -d ' ' -f1) #use apparent size to display
correct size of NAS! otherwise ~200 MB will show as ~3.5 GB

#mail
printf "[%DTS] rsync-Backup of $(hostname -f) was done! \nSize:"$SIZE"" | mail -s "$(hostname
-f)" target@mail.address

```

Remote Backup

Additionally a remote host is copying the backup files by pulling with rsync overSSH.

Cleaning old InfluxDB dumps

/etc/cron.d/clean-influx-bup

```
SHELL=/bin/bash
PATH=/usr/lib/sysstat:/usr/sbin:/usr/sbin:/usr/bin:/sbin:/bin
0 21 * * * root find /mnt/external_usb32g/db/ -mtime +14 -exec rm {} \; > /dev/null
```

Troubleshooting Postfix (Restart Postfix)

Because sometimes mailq is not emptied we are going to force sending out mails every 5 minutes. Sometimes the message "postqueue: warning: Mail system is down -- accessing queue directly" is shown when typing command "mailq".

/etc/cron.d/restartpostfix

```
SHELL=/bin/bash
PATH=/usr/lib/sysstat:/usr/sbin:/usr/sbin:/usr/bin:/sbin:/bin
*/5 * * * * root systemctl restart postfix
*/5 * * * * root postsuper -r ALL > /dev/null 2>&1
```

Send dmesg log by mail

/etc# cat rc.local

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
```

```
# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
    printf "My IP address is %s\n" "$_IP"
fi

# send dmesg log by mail once per reboot
dmesg | mail -s "dmesg log from session" target@mail.address

exit 0
```

Switch off LED stripes and spot from Trikarus

Saves energy and prolongs the life of the LEDs

/etc/cron.d/led_off_trikarus

```
# this script switches off the LED stripes and spot from Trikarus at 18:00 every day
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
0 17 * * *      root  python3.7 /opt/gpio/ledspot-off.py
```

Switch on LED stripes and spot from Trikarus

See <https://web.saechsisches-industriemuseum.com/chemnitz/infothek/oeffnungszeiten-preise-anfahrt/oeffnungszeiten.html>

This script switches on the LEDs once per hour. It should not autotrigger every minute because the userdefined request to turn them off should be respected too (from bash or from Repetier Server UI)!

/etc/cron.d/led_on_trikarus

```
# this script switches on the LED stripes and spot from Trikarus
#"At every minute past every hour from 9 through 17 on Tuesday, Wednesday, Thursday, Friday, Sat
SHELL=/bin/sh
```

```
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
0 9-17 * * 2,3,4,5,6,7 root python3.7 /opt/gpio/ledspot-on.py
```

Hourly calendar event upload

As we need to switch to Trikanet when running the printer locally, we use the predetermined calendar dates, parsed by main server, then uploaded to hangprinter raspberry pi. From this it get's synced to the Freifunk Router

/etc/cron.d/hangprinter-shows

```
SHELL=/bin/bash
PATH=/usr/lib/sysstat:/usr/sbin:/usr/sbin:/usr/bin:/sbin:/bin
#hourly try to upload calendar data from hangprinterbackup account. that file gets created on ma
0 * * * * root scp -v -6 -i /root/.ssh/c-MaschinenBoom-FLC /home/hangprinterbackup/sonstiges.crc
```

See [FCL:Diverse Scripts](#) on how Neycerha pushed calendar data to hangprinter.

Version #1

Erstellt: 2026-06-08 14:21:37 CEST von Mario Voigt

Zuletzt aktualisiert: 2026-06-08 14:23:49 CEST von Mario Voigt