

LoxBerry und RAM-Disk

LoxBerry verwendet zur Reduktion der Schreibzugriffe für sehr viele Systemdateien (hauptsächlich Log-Dateien) als Dateisystem eine RAM-Disk. Jede eingehängte RAM-Disk wird mit einem Maximalwert konfiguriert, der jedoch nicht sofort belegt wird. Nur der tatsächlich von Dateien auf der RAM-Disk belegte Speicher wird im RAM allokiert.

Beim Booten des LoxBerry werden notwendige Ordner für Systemdienste und Plugins auf der RAM-Disk angelegt, bevor diese gestartet werden.

RAM-Disk ab LoxBerry 2.0

LoxBerry verwendet - abhängig von der Hardware - zwei unterschiedliche Treiber für die RAM-Disk:

Hardware	RAM-Disk-Typ	Maximale Größe
Raspberry Pi 1/2/Zero	tmpfs	200 MB
Raspberry 3/3+/4	zram	415 (200) MB
Andere Hardware, VMs	zram	415 (200) MB

Der zram-Treiber verwendet Kompression bei der Speicherung auf der RAM-Disk. Der Wert in Klammern repräsentiert dabei das maximale Speicherlimit für die komprimierten Daten, während der andere Wert die Volumegröße darstellt.

Die langsameren Raspberry's verwenden weiterhin tmpfs, da die Kompression von zram auf diesen Geräten die Performance deutlich beeinflussen würde.

Die angegebenen Werte sind Standardwerte. In der Datei `/opt/loxberry/config/system/general.json` können die Verwendung von zram/tmpfs, die Speichergrößen und der Kompressionsalgorithmus von zram manuell angepasst werden. Dafür muss `Log2ram.Manualconfigured = true` gesetzt werden, dann werden die Werte des `Log2ram`-Objektes beim nächsten Start übernommen. So kann z.B. auch auf dem Pi 1 zram aktiviert werden, oder umgekehrt, zram am Pi3/4 deaktiviert.

Binds

Typ	Quelle	Ziel	
mount	tmpfs oder zram	<code>/opt/loxberry/log/ramlog</code> (= <code>\$RAMLOG</code>)	Root-Verzeichnis der RAM-Disk
bind	<code>/var/log</code>	<code>\$RAM_LOG/var/log</code>	System-Logverzeichnis
bind	<code>/tmp</code>	<code>\$RAM_LOG/tmp</code>	System-Temp-Verzeichnis
bind	<code>/var/tmp</code>	<code>\$RAM_LOG/var/tmp</code>	System-Temp-Verzeichnis (persistiert)
bind	<code>\$LBHOMEDIR/log/plugin</code>	<code>\$RAM_LOG/log/plugins</code>	Plugin-Logverzeichnisse
bind	<code>\$LBHOMEDIR/log/system_tmpfs</code>	<code>\$RAM_LOG/log/system_tmpfs</code>	LoxBerry-Core-Logverzeichnisse

Quellcode der

RAM-
Disk-

Initialisierung: <https://github.com/mschlenstedt/Loxberry/blob/master/sbin/createtmpfsfoldersinit.sh>

RAM-Disk ab LoxBerry V1.2.5

tmpfs Mountpoints

tmpfs-Volumes

Mountpoint	Max-Size	Pi1	Pi2/3/3+
/dev/shm	50% des RAM's	~ 232 MB	~ 464 MB

Der Mount von /dev/shm ist nicht in der fstab aufgelistet, da dies von systemd verwaltet und gemounted wird.

Binds

In die genannte RAM-Disk /dev/shm bindet LoxBerry mehrere Verzeichnisse. Dies passiert während des Startups von LoxBerry in createtmpfsfoldersinit.sh

<https://github.com/mschlenstedt/Loxberry/blob/5415ca5f75327f238deed5d71610d667684dcb55/sbin/createtmpfsfoldersinit.sh#L48>

Darunter fallen alle Log-Verzeichnisse der Plugins.

Cleanup von Log-Dateien

LoxBerry führt ein stündliches, automatisches Cleanup von Logdateien des Systems und von Plugins durch. Dies betrifft alle Logdateien, nicht nur jene, die über das Logging SDK erstellt wurden. Dieses Cleanup läuft in mehreren Schritten mit zunehmender "Aggressivität":

Stage	Filter	Durchgeführte Aktion
<u>Stage 1:</u> Jede Stunde ohne Einschränkung	Alle *.log Dateien über 1 MB Größe	* Zippen als *.log.gz * Löschen des Logfiles
	Alle *.log Dateien nicht geändert seit über 30 Tagen	* Zippen als *.log.gz * Löschen des Logfiles
	Alle *.gz Dateien nicht geändert seit über 60 Tagen	* Löschen des .gz Files
<u>Stage 2:</u> Alle tmpfs Laufwerke mit Speicherplatz unter 25%	Alle *.gz Dateien über 1 MB Größe	* Löschen des .gz Files

Stage	Filter	Durchgeführte Aktion
Stage 3: Alle tmpfs Laufwerke mit Speicherplatz unter 25%	Alle *.gz Dateien	* Löschen des .gz Files
Stage 4: Alle tmpfs Laufwerke mit Speicherplatz unter 5%	Alle *.log Dateien	* Löschen des Logfiles
Stage 5: Logging SDK	Anzahl der Logfiles der PACKAGE+NAME-Kombination über 20 Stk	* Löschen des Logfiles

Healthcheck / Selbsttest

Die Prüfung auf freien Speicherplatz auf der RAM-Disk ist Teil von LoxBerrys [Healthcheck](#) (Selbsttest). Wenn zu wenig oder kein Speicherplatz mehr verfügbar ist, meldet der Check einen Fehler. Tritt dieser Fehler regelmäßig auf, ist zu prüfen, welche Daten dort gespeichert sind (dies könnte beispielsweise durch ein Plugin ausgelöst sein).

From:

<https://wiki.loxberry.de/> - **LoxBerry Wiki - BEYOND THE LIMITS**

Permanent link:

https://wiki.loxberry.de/haufig_gestellte_fragen_faq/loxberry_und_ram_disk

Last update: **2022/09/10 12:18**