

Partition aufs maximum vergrößern wenn die Festplatte vergrößert wurde

Beschreibung:

Es kann mal vor kommen, das eine Festplatte z.b in einer VM vergrößert wird. Leider wird das Dateisystem nicht mit angepasst.

Hier ein Script das die macht.

Dies Script hat allerdings Pakete Voraussetzungen die noch installiert werden müssen.

Voraussetzungen:

```
apt-get install bc e2fsprogs parted util-linux cloud-utils
```

Das Script:

```
nano /root/extend_to_max.sh
```

Inhalt

```
#!/bin/bash
#
# extend_to_max.sh - Erweitert eine Partition (und das Dateisystem)
#                   auf den maximal verfügbaren freien Speicher.
#
# Benutzung:
#   ./extend_to_max.sh -d <Partition>
#
```

```
# Beispiel:
# ./extend_to_max.sh -d /dev/sda2
#
# ACHTUNG: Vor Änderungen unbedingt ein vollständiges Backup anfertigen!
#

print_help() {
    echo "Usage: $0 -d <Partition>"
    echo "  -d Zielpartition (z.B. /dev/sda2)"
}

# Parameter prüfen
if [ "$#" -eq 0 ]; then
    print_help
    exit 1
fi

while getopts "d:h" opt; do
    case $opt in
        d)
            DEVICE="$OPTARG"
            ;;
        h)
            print_help
            exit 0
            ;;
        ?)
            print_help
            exit 1
            ;;
    esac
done

if [ -z "$DEVICE" ]; then
    echo "Fehler: Kein Gerät angegeben!"
    print_help
    exit 1
fi

# Root-Prüfung
```

```

if [ "$(id -u)" -ne 0 ]; then
    echo "Dieses Script muss als root ausgeführt werden."
    exit 1
fi

echo "Erweitere Partition $DEVICE auf maximal verfügbaren Speicher..."

# Bestimme den übergeordneten Datenträger und die Partitionsnummer.
PARENT=$(lsblk -no pkname "$DEVICE" 2>/dev/null)
if [ -n "$PARENT" ]; then
    DISK="/dev/$PARENT"
else
    DISK=$(echo "$DEVICE" | sed 's/[0-9]*$//')
fi

PART_NUM=$(echo "$DEVICE" | sed 's/.*\([0-9]\+\)$\1/')
echo "Verwende Datenträger: $DISK, Partition: $PART_NUM"

# Starte die Erweiterung mit growpart.
# (growpart erweitert standardmäßig bis zum maximal verfügbaren freien Platz)
echo "Starte growpart..."
growpart "$DISK" "$PART_NUM"
if [ $? -ne 0 ]; then
    echo "Fehler: growpart konnte die Partition nicht erweitern."
    exit 1
fi

# Erweitere anschließend das Dateisystem (hier wird ext2/3/4 vorausgesetzt).
echo "Passe das Dateisystem an ($DEVICE) an..."
resize2fs "$DEVICE"
if [ $? -ne 0 ]; then
    echo "Fehler: resize2fs konnte das Dateisystem nicht erweitern."
    exit 1
fi

echo "Partition und Dateisystem wurden erfolgreich auf den maximal verfügbaren Speicher erweitert."
exit 0

```

Script Ausführbar machen

```
chmod +x /root/extend_to_max.sh
```

Erklärung:

1. Parameterverarbeitung:

- `-d <Partition>`: Gibt die Partition an, die vergrößert werden soll (z.B. `/dev/sda1`).

2. Überprüfung der Partition:

- Das Skript überprüft, ob die angegebene Partition existiert.

3. Vergrößern der Partition:

- Das Skript verwendet `growpart`, um die Partition zu vergrößern.

4. Vergrößern des Dateisystems:

- Das Skript verwendet `resize2fs`, um das Dateisystem auf der Partition zu vergrößern.

Hilfe anzeigen:

```
./extend_to_max.sh
```

Vergrößerung:

```
./extend_to_max.sh -d /dev/sda1
```

Version #10

Erstellt: 2 Februar 2025 08:08:18 von Admin

Zuletzt aktualisiert: 2 Februar 2025 10:13:12 von Admin