YTVI Autostart in Linux

Autostart-Dateien und Mechanismen unter Debian/Ubuntu

Übersicht

Datei / Mechanismus	Startzeitpunkt	Benutzerbereich	Kontext (Konsole oder X)	Ruft andere Skripte auf	Besonderheiten und Hinweise	Beispiele für typische Einträge
`/etc/profile`	Login-Shell	Global (Alle Benutzer)	Konsole (Bash/Zsh)	Ruft `/etc/profile.d/*` auf	Diese Datei wird bei der Anmeldung eines Benutzers (Login-Shell) ausgeführt. Sie wird bei Terminal-Anmeldung oder SSH verwendet, aber nicht bei X-Session-Starts. Standardmäßig für systemweite Umgebungsvariablen geeignet.	Setzen von systemweiten Umgebungsvariablen (`export PATH=…`)
`~/.profile`	Login-Shell	Spezifisch für den Benutzer	Konsole und X	Ruft `~/.bash_profile` oder `~/.bashrc` auf (abhängig von der Konfiguration)	Wird bei der Anmeldung mit einer Login-Shell oder bei X-Anmeldung verwendet. Wird oft verwendet, um benutzerspezifische Umgebungsvariablen zu setzen. In Bash ist `.profile` die Standarddatei für Login-Shells. Wird von Bash, Dash und möglicherweise auch X-Session gelesen.	* Setzen von benutzerspezifischen Aliases und Umgebungsvariablen * Starten von Programmen (eval $(ssh-agent -s))
`~/.bashrc`	Login-Shell und Nicht-Login-Shell	Spezifisch für den Benutzer	Konsole (Bash)	Wird durch `~/.profile` aufgerufen, wenn es eine Login-Shell ist	Diese Datei wird bei jedem Start einer neuen nicht-Login-Bash-Shell ausgeführt. Üblicherweise für Terminal-Fenster und interaktive Shells verwendet, jedoch nicht für grafische Sitzungen oder Login-Shells.	Aliases (`alias ll='ls -la'`) und benutzerspezifische Funktionen
`~/.zshrc`	Nicht-Login-Shell	Spezifisch für den Benutzer	Konsole (Zsh)	-	Ähnlich wie `.bashrc`, jedoch für die Zsh-Shell. Wird jedes Mal ausgeführt, wenn eine Zsh-Instanz gestartet wird (nicht-Login-Shell).	Aliases und benutzerspezifische Zsh-Einstellungen
`/etc/rc.local`	Systemstart (letzte Phase)	Global (Alle Benutzer)	Nicht spezifiziert (Systemebene)	-	Diese Datei wird in der letzten Phase des Systemstarts ausgeführt. Sie eignet sich, um systemweite Dienste oder Befehle auszuführen. Seit neueren Versionen von Debian/Ubuntu kann es sein, dass `/etc/rc.local` standardmäßig deaktiviert ist. Man kann sie manuell aktivieren. Wird als `root` ausgeführt, da es Teil des Systemstarts ist.	Starten von benutzerdefinierten Diensten (`/usr/bin/mein_dienst &`)
`/etc/X11/Xsession`	Start der X-Sitzung	Global (Alle Benutzer)	X (Grafische Umgebung)	Ruft benutzerspezifische `~/.xsession` auf (falls vorhanden)	Diese Datei wird verwendet, um die X-Session (Grafiksession) zu initialisieren. Sie ist ein zentraler Punkt für die Konfiguration der X-Sitzung und wird vor der Ausführung von Desktop-Umgebungen (wie GNOME oder KDE) aufgerufen.	* Setzen von Umgebungsvariablen für X (`export GTK_THEME=…`) * Deaktivieren des Bildschirmschoners (`xset s off -dpms`)
`~/.xinitrc`	Start von X via `startx`	Spezifisch für den Benutzer	X (Grafische Umgebung)	-	Diese Datei wird verwendet, wenn ein Benutzer den X-Server mit dem Befehl `startx` startet. Typischerweise für Benutzer, die keine Display-Manager wie `gdm` oder `lightdm` verwenden.	* Starten des Fenstermanagers (`exec i3`) * Deaktivieren des Bildschirmschoners (xset s off -dpms)

Datei / Mechanismus	Startzeitpunkt	Benutzerbereich	Kontext (Konsole oder X)	Ruft andere Skripte auf	Besonderheiten und Hinweise	Beispiele für typische Einträge
`~/.xsession`	Login über Display-Manager	Spezifisch für den Benutzer	X (Grafische Umgebung)	-	Diese Datei wird ausgeführt, wenn der Benutzer sich über einen Display-Manager (z.B. `gdm` oder `lightdm`) anmeldet. Sie ermöglicht benutzerspezifische Konfigurationen für die grafische Session.	Setzen von benutzerspezifischen Umgebungsvariablen
`/etc/init.d/* (SysV)`	Systemstart	Global (Alle Benutzer)	Nicht spezifiziert (Systemebene)	-	Wird verwendet, um systemweite Dienste zu starten. Veraltet zugunsten von systemd, aber immer noch in Verwendung bei älteren Systemen.	Starten eines systemweiten Dienstes (`/etc/init.d/apache2 start`)
`systemd Units (/etc/systemd/system/*)`	Systemstart / Benutzeranmeldung	Global oder Benutzerabhängig (je nach Konfiguration)	Nicht spezifiziert (Systemebene)	Kann andere Units oder Skripte per `ExecStart` aufrufen	Systemd ist der aktuelle Standard für den Dienststart auf modernen Debian/Ubuntu-Systemen. Man kann eigene Units definieren, die Programme oder Skripte bei Systemstart oder bei Benutzeranmeldungen ausführen.	Definieren eines Dienstes (`ExecStart=/usr/bin/mein_dienst`)
`~/.config/autostart/*.desktop`	Login der grafischen Session	Spezifisch für den Benutzer	X (Grafische Umgebung)	-	Diese Methode wird verwendet, um Anwendungen automatisch beim Start der grafischen Desktop-Umgebung (wie GNOME oder KDE) zu starten. Dateien im `.desktop`-Format definieren, welche Programme gestartet werden sollen.	Starten einer Anwendung (`Exec=/usr/bin/firefox`)
`/etc/xdg/autostart/*.desktop`	Login der grafischen Session	Global (Alle Benutzer)	X (Grafische Umgebung)	-	Ähnlich wie `~/.config/autostart`, aber systemweit für alle Benutzer. Diese Methode wird ebenfalls von Desktop-Umgebungen genutzt, um Programme automatisch zu starten.	Starten einer systemweiten Anwendung (`Exec=/usr/bin/gnome-shell`)

Hinweise

1. **Unterschied Login-Shell vs. Nicht-Login-Shell**:

    1. Eine Login-Shell ist die erste Shell, die gestartet wird, wenn ein Benutzer sich im System anmeldet (z.B. über SSH oder die Konsole).
    2. Eine Nicht-Login-Shell wird beispielsweise durch das Starten eines Terminal-Fensters aus einer bestehenden Sitzung heraus geöffnet.

2. **Global vs. Benutzerspezifisch**:

    1. Dateien unter `/etc/` sind normalerweise global, d.h. sie gelten für alle Benutzer.
    2. Dateien in den Home-Verzeichnissen (`~/.bashrc`, `~/.profile` etc.) gelten nur für den spezifischen Benutzer.

3. **Grafische Umgebung vs. Konsole**:

    1. Einige dieser Dateien sind speziell für die grafische Umgebung (X) gedacht (`~/.xinitrc`, `~/.xsession`, `~/.config/autostart/`).
    2. Andere wiederum sind speziell für Shell-Sitzungen (Login und Nicht-Login) gedacht (`~/.bashrc`, `/etc/profile`).

4. **Systemd für modernere Verwaltung**:

    1. Neuere Versionen von Debian/Ubuntu setzen hauptsächlich auf `systemd`, um Startprozesse zu

verwalten. Man kann dafür eigene “Service”-Dateien erstellen, um Programme beim Systemstart oder bei Benutzeranmeldungen auszuführen.

5. **.desktop-Dateien für die grafische Umgebung**:

   1. `.desktop`-Dateien, die in den Autostart-Ordnern (`~/.config/autostart/` oder `/etc/xdg/autostart/`) gespeichert werden, dienen dazu, Anwendungen automatisch beim Login in einer grafischen Session zu starten. Diese Methode ist besonders praktisch für Anwendungen mit GUI.

zeitlichen Abfolge

```
Systemstart
|
|-- `/etc/init.d/*` (SysV) oder `systemd` Units (`/etc/systemd/system/*`)
|   |
|   |-- `/etc/rc.local` (letzte Phase des Systemstarts)
|
Benutzeranmeldung (Login-Shell)
|
|-- `/etc/profile`
|   |
|   |-- `/etc/profile.d/*`
|
|-- `~/.profile`
    |
    |-- Ruft ggf. `~/.bash_profile` oder `~/.bashrc` auf
```

```
Start von nicht-Login-Shell (z.B. neues Terminal)
|
|-- `~/.bashrc` (für Bash)
|-- `~/.zshrc` (für Zsh)
```

```
Start der grafischen Umgebung (X-Session)
|
|-- `/etc/X11/Xsession`
|   |
|   |-- Ruft ggf. `~/.xsession` auf
|
|-- `~/.xinitrc` (bei Start mit `startx`)
|-- `~/.xsession` (bei Anmeldung über Display-Manager)
|
|-- `/etc/xdg/autostart/*.desktop` (systemweit für alle Benutzer)
|-- `~/.config/autostart/*.desktop` (benutzerspezifisch)
```

Beispiel-Szenarien

1. **Skript systemweit beim Booten ausführen**:

1. Verwenden Sie `/etc/rc.local` oder erstellen Sie eine `systemd` Unit, um ein Skript auszuführen, bevor Benutzer sich anmelden.

2. **Umgebungsvariablen für alle Benutzer setzen**:

   1. Verwenden Sie `/etc/profile`, wenn es systemweite Variablen betrifft, oder `~/.profile` für spezifische Benutzer.

3. **Programm für grafische Sitzung eines Benutzers starten**:

   1. Platzieren Sie eine `.desktop`-Datei in `~/.config/autostart/`, um sicherzustellen, dass das Programm automatisch beim Anmelden an die grafische Benutzeroberfläche gestartet wird.

Autostart Log

- in /home/pi eine Datei anlegen
  `nano /home/pi/log.sh`

```
#!/bin/bash

# Log-Datei
LOG_FILE="/tmp/startup_log.txt"

# Überprüfe, ob das Skript als Root ausgeführt wird
if [ "$EUID" -eq 0 ]; then
  # Überprüfe, ob die Logdatei existiert, falls nicht, erstelle sie und
setze Berechtigungen
  if [ ! -f "$LOG_FILE" ]; then
    touch "$LOG_FILE"
    chmod 666 "$LOG_FILE"
    echo "$(date '+%Y-%m-%d %H:%M:%S.%3N') - Log file created and
permissions set to 666 by root" >> "$LOG_FILE"
  fi
fi

# Überprüfe, ob ein Parameter übergeben wurde
if [ -z "$1" ]; then
  echo "Usage: $0 'log message'"
  exit 1
fi

# Log-Daten sammeln
TIMESTAMP=$(date '+%Y-%m-%d %H:%M:%S.%3N')
USER=$(whoami)
HOSTNAME=$(hostname)
IP_ADDRESS=$(hostname -I)
LOAD=$(uptime | awk '{print $8,$9,$10}')  # CPU-Last
MEMORY=$(free -m | awk 'NR==2{printf "%s/%sMB (%.2f%%)",
$3,$2,$3*100/$2 }')  # RAM-Auslastung
PID=$$
```

```
#PPID=$PPID  # Elternprozess-ID
GROUPS=$(groups $USER)
OS_KERNEL=$(uname -sr)
UPTIME=$(uptime -p)
DISK_USAGE=$(df -h / | awk 'NR==2 {print $5}')
WORK_DIR=$(pwd)
TTY=$(tty)

# Log-Eintrag schreiben
echo "$TIMESTAMP - $USER@${HOSTNAME} - Mem: $MEMORY - PPID: $PPID -
WDir: $WORK_DIR - TTY: $TTY > $1" >> $LOG_FILE
```

- sudo chmod 666 /home/pi/log.sh
- Ergebnis

```
root@FriemelKiste:~# cat /tmp/startup_log.txt
2024-10-03 23:33:36.740 - Log file created and permissions set to 666
by root
2024-10-03 23:33:36.744 - root@FriemelKiste - Mem: 93/973MB (9.56%) -
PPID: 1 - WDir: / - TTY: not a tty > Service Start
2024-10-03 23:33:36.740 - root@FriemelKiste - Mem: 93/973MB (9.56%) -
PPID: 2320 - WDir: /root - TTY: not a tty > Reboot aus Cron -e
2024-10-03 23:33:52.212 - root@FriemelKiste - Mem: 102/973MB (10.48%) -
PPID: 2552 - WDir: / - TTY: not a tty > /etc/rc.local
2024-10-03 23:33:53.734 - pi@FriemelKiste - Mem: 105/973MB (10.79%) -
PPID: 2622 - WDir: /home/pi - TTY: /dev/tty1 > /etc/profile -> All User
2024-10-03 23:33:54.082 - pi@FriemelKiste - Mem: 107/973MB (11.00%) -
PPID: 2622 - WDir: /home/pi - TTY: /dev/tty1 > /home/pi/.bashrc -> pi
2024-10-03 23:33:58.945 - root@FriemelKiste - Mem: 130/973MB (13.36%) -
PPID: 2781 - WDir: /root - TTY: /dev/pts/0 > /etc/profile -> All User
2024-10-03 23:33:59.035 - root@FriemelKiste - Mem: 132/973MB (13.57%) -
PPID: 2781 - WDir: /root - TTY: /dev/pts/0 > ROOT /.bashrc
2024-10-03 23:33:59.116 - root@FriemelKiste - Mem: 136/973MB (13.98%) -
PPID: 2781 - WDir: /root - TTY: /dev/pts/0 > .profile -> root
2024-10-03 23:34:04.508 - pi@FriemelKiste - Mem: 213/973MB (21.89%) -
PPID: 3023 - WDir: /home/pi - TTY: /dev/pts/1 > /home/pi/.bashrc -> pi
2024-10-03 23:35:13.225 - pi@FriemelKiste - Mem: 217/973MB (22.30%) -
PPID: 3105 - WDir: /home/pi - TTY: /dev/pts/2 > /etc/profile -> All
User
2024-10-03 23:35:13.304 - pi@FriemelKiste - Mem: 222/973MB (22.82%) -
PPID: 3105 - WDir: /home/pi - TTY: /dev/pts/2 > /home/pi/.bashrc -> pi
2024-10-03 23:35:13.380 - pi@FriemelKiste - Mem: 234/973MB (24.05%) -
PPID: 3105 - WDir: /home/pi - TTY: /dev/pts/2 > .profile -> Pi
2024-10-03 23:35:33.964 - pi@FriemelKiste - Mem: 130/973MB (13.36%) -
PPID: 2622 - WDir: /home/pi - TTY: /dev/tty1 > .profile -> Pi
2024-10-03 23:42:53.668 - root@FriemelKiste - Mem: 123/973MB (12.64%) -
PPID: 3374 - WDir: /root - TTY: /dev/pts/1 > /etc/profile -> All User
2024-10-03 23:42:53.744 - root@FriemelKiste - Mem: 123/973MB (12.64%) -
PPID: 3374 - WDir: /root - TTY: /dev/pts/1 > ROOT /.bashrc
2024-10-03 23:42:53.816 - root@FriemelKiste - Mem: 135/973MB (13.87%) -
PPID: 3374 - WDir: /root - TTY: /dev/pts/1 > .profile -> root
```

```
2024-10-03 23:42:58.707 - root@FriemelKiste - Mem: 221/973MB (22.71%) -
PPID: 3654 - WDir: /root - TTY: /dev/pts/2 > ROOT /.bashrc
```

From:
https://www.drklipper.de/ - **Dr. Klipper Wiki**

Permanent link:
https://www.drklipper.de/doku.php?id=linux:ytvi_autostart_in_linux&rev=1728798700

Last update: **2024/10/13 07:51**