ICT

§9 Services, Storage & Internals

Harald Schilly
harald.schilly@univie.ac.at

Service / Daemon (1)

Ein Service, oder Daemon, ist ein Prozess der im Hintergrund läuft.
Beispiel: Webserver, Datenbank, Cron, ...
Eigenschaften:

- Startet automatisch beim Booten.
- Isoliert: unter eigenem Namen/Gruppe.
- Läuft durchgehend im Hintergrund.

Init-Skripte bzw. Utilities wie Upstart steuern, wann welcher Daemon gestartet wird.
Upstart (Ubuntu)

- /etc/init - Konfigurationsfiles, siehe $ `man 5 init`
- /etc/init.d/* - "alte" System-V Skripts, von Upstart aufgerufen.

Das Administrationstool "service" kann Daemons starten oder stoppen indem es entsprechende Skripte aufruft.
Mehr: `man service, man upstart, man upstart-events, man startup,…`

Service / Daemon (2)

Init-Skripte werden mit update-rc.d konfiguriert. Sie haben den Syntax:
`/etc/rc[runlevel].d/[SK][NN][name]`

Runlevels werden von Upstart als Events ausgeführt (`man 7 runlevel`), es werden dann die Skripte in der Reihenfolge ihrer Sequenznummer NN abgearbeitet.
S heißt Start, K heißt Kill.

Daumenregel: Seqzenznummer der Stop-Einträge sollte 100 minus der Sequenznummer der Starteinträge sein - damit es in umgekehrter Reihenfolge abläuft.

z.B.: Installation eines neuen init-Skripts:

```
$ update-rc.d <skript> defaults
```

LVM (1)

LVM (Logical Volume Manager) erlaubt es, mehrere Festplatten zusammenzufassen und übergreifende Dateisysteme zu erstellen.

Die dabei entstehenden Dateisysteme sind üblicherweise weit größer als die der einzelnen Festplatten, es kann dynamisch repartitioniert werden, und Änderungen an der Konfiguration der Festplatten können dynamisch in die Größen der Dateisysteme einfließen.

Auch können "Snapshots" für konsistente Backups gemacht werden.

Links: lvm2, lvm howto, wikipedia (en), webmanual

LVM/PV (2)

- **PV**: "Physical Volume". Das sind die real existierenden Festplatten (block devices). Vor deren Verwendung müssen sie mit diesem Kommando initialisiert werden.
  - `pvcreate /dev/sd…`
  - `pvscan` ... sucht in /dev nach möglichen Festplatten
  - `pvdisplay`

LVM/VG (3)

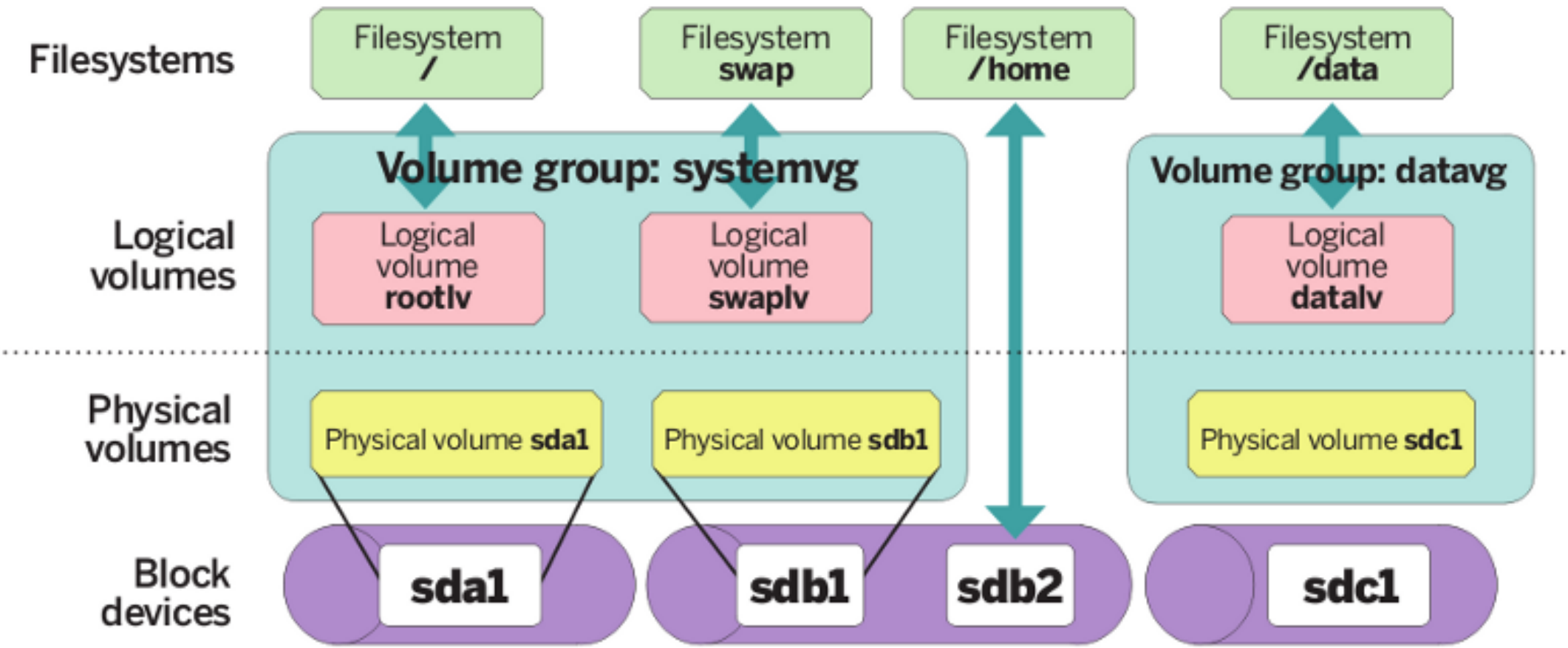
- **VG**: "Volume Group": Mehrere dieser PVs werden zu VGs zusammengefasst. Eine VG kann später modifiziert werden und ist die die Ausgangsbasis für die spätere Partitionierung.
  - `vgcreate <ID Name> /dev/sd... /dev/sd... ...`
  - `vgdisplay`
  - `vgs` ... Statusinformationen
  - `vgremove` vs. `vgcreate`
  - `vgreduce` vs. `vgextend` ... entfernt oder fügt Festplatten hinzu. Es wird dabei auch die Größe geändert (Späteres anpassen des eigentlichen Dateisystems nicht vergessen!)

LVM/LV (4)

- **LV**: "Logical Volume": Das sind die "Partitionen", die in den VGs erstellt werden.
  - `lvcreate -L 300M -n LVLoop0 VGLoop`
    (erzeugt z.B.: `/dev/VGLoop/LVLoop0`)
  - `lvdisplay`
  - `lvs` ... Statusinformationen
  - `lvchange` ... Attribute ändern

... Anschließend kann man wie gehabt mit `mkfs.*` die LV formatieren um ein Dateisystem zu bekommen, das man anschließend mounten kann. Achtung: Es sollte ein Dateisystem sein, dessen Größe man später ändern kann! (`ext2/3/4` mit `resize2fs`)

LVM (5)



Credits: http://www.tuxradar.com/content/lvm-made-easy

(Software) RAID

(Software) RAID (Redundant Array of Independent Disks)
Das ist eine Technik, um Datenverlust bei einem Ausfall einer Festplatte zu vermeiden. Ein Managementprogramm erzeugt eine virtuelle Festplatte (zB /dev/md0) und spricht damit mehrere Festplatten an. /dev/md0 kann dann normal verwendet werden (Partitionieren + Mounten)

- RAID 0: Inhalte nur aneinandergereiht, keine Sicherheit!
- **RAID 1**: (Mirror): Der Inhalt einer Festplatte wird 1:1 auf anderen gespiegelt. Fällt eine aus, hat die andere denselben Inhalt.
  - Unklar was zu tun ist wenn Inhalte unterschiedlich sind!
- **RAID 5**: Daten mit Prüfsummen sind auf mehreren Platten verteilt, Ausfall nur einer Disk schaded nicht. Wozu werden die Prüfsummen im Unterschied zu RAID 1 verwendet?
- RAID 6: Wie 5, aber die Prüfsummen sind immer doppelt vorhanden. Es dürfen sogar 2 Platten ausfallen ohne dass es Datenverlust gibt.
- Hybride: RAID 1+0 oder RAID 0+1

Link: RAID

MDADM

MDADM (Multiple Disk Administration) richtet Software RAIDs ein.
Konfiguration:

```
$ mdadm --create /dev/md0 --level=mirror --raid-devices=2 /dev/sda1 [2]

$ mdadm --create /dev/md1 --level=5 --raid-devices=3 /dev/[1] [2] [3]
```

... nach diesem "create" gibt es ein Device /dev/md[N] das in "Wahrheit" die angebenen anderen /dev/sd[..] oder /dev/hd[..] gleichzeitig anspricht.

`$ mdadm --detail /dev/md0` ... Information
`$ mdadm --stop /dev/md0` ... Deaktivierung (`--remove /dev/md0`)

Zeile in mdadm.conf eintragen (wie /dev/fstab), zB:

```
mdadm -Es | grep md0 >> /dev/mdadm/mdadm.conf
```

Mit mdadm --fail, --add und --remove ist es möglich, (defekte) Festplatten während des laufenden Betriebs zu wechseln (zB wenn mehrere über USB angeschlossen sind) ohne dass es einen Datenverlust gibt.

MDADM + LVM

Kombination beider Ansätze möglich:

1. Mehrere Festplatten mittels mdadm redundant zusammenfassen (zB RAID 5)
2. Dieses /dev/md[N] Gerät als PV in LVM definieren.
3. Eine VG mit diesem PV anlegen.
4. Mehrere LV mit Dateisystemen in dieser VG anlegen, die dann ihre Größe dynamisch ändern können, Migration, etc.

/proc

/proc ist ein System-Verzeichnis und erlaubt Einblicke in den momentanen Zustand des Betriebssystems Linux und ggf. ist es möglich Parameter anzupassen.

- `cat /proc/../key` ... ließt Daten aus
- `echo "string" > /proc/../key` ... setzt eine Einstellung

- `/proc/filesystems` - vom Kernel unterstütze Dateisysteme
- `/proc/loadavg` - Systemauslastung:
  CPU Auslastung in den letzten 1, 5, und 10 Minuten; aktive/alle Prozesse, letzte aktive PID.

```
$ cat /proc/loadavg
0.18 0.19 0.22 1/366 16356
```

/proc (2)

Informationen über Interrupts, ID, für jede CPU, Emfänger:

$ cat /proc/interrupts

```
           CPU0       CPU1
  0:       4362       3789   IO-APIC-edge      timer
  1:          2          0   IO-APIC-edge      i8042
  4:          1          1   IO-APIC-edge
  6:          3          0   IO-APIC-edge      floppy
  7:          0          0   IO-APIC-edge      parport0
  8:          0          1   IO-APIC-edge      rtc0
  9:          0          0   IO-APIC-fasteoi   acpi
 12:          3          1   IO-APIC-edge      i8042
 16:    2960744    2574485   IO-APIC-fasteoi   ehci_hcd:usb2, uhci_hcd:usb3, uhci_hcd:usb6, HDA Intel
 17:          0          1   IO-APIC-fasteoi   uhci_hcd:usb4, uhci_hcd:usb7
 18:          0          0   IO-APIC-fasteoi   ehci_hcd:usb1, uhci_hcd:usb5, uhci_hcd:usb8
...
...
```

/proc (3)

/proc/locks - listet Dateien mit "locks" auf. Das sind Dateien, oder Bereiche in Dateien, auf die Prozesse exklusive (schreib/lese) Rechte haben. Es soll verhindert werden, dass zwei Prozesse gleichzeitig dieselbe Datei verändern und es dann zu Unstimmigkeiten kommt.

Shellbefehl: **`flock, lockfile-*`**
Beispiel: führe folgende zwei Befehle parallel aus:

```
$ flock -x somefile sleep 10
$ flock -x [-n] somefile echo "hi"
```

`-n` ... non-blocking, d.h. es wird nicht auf Freigabe gewartet

Einträge: ID, Typ, advisory/mandatory, read/write, PID, Identifikation für Datei (gerät-major:minor:Inode-Zahl),...

```
7: FLOCK  ADVISORY  WRITE 3467 00:12:9819 0 EOF
```

- major/minor Gerät in `/proc/partitions`

/proc (3.1)

Shellbefehl: **lockfile-***

```
#!/usr/bin/env bash
lockfile-create somefile
lockfile-touch somefile &
# Save the PID of the lockfile-touch process
BADGER="$!"
echo $BADGER > somefile
sleep 10 # do something
kill "${BADGER}"
lockfile-remove somefile
```

Während das Skript läuft (insbesondere lockfile-touch), ist es nicht möglich erneut ein lockfile-create auf genau diese Datei auszuführen. Die Ausführung wird angehalten bis dies möglich ist (wait)
oder bricht ab. Damit wird zB sichergestellt, dass ein Skript
nur exakt einmal aufgerufen wird.

/proc (4)

$ cat /proc/meminfo #Informationen über Memory

```
MemTotal:        1894608 kB
MemFree:           68508 kB
Buffers:            4928 kB
Cached:           191424 kB
SwapCached:        65408 kB
Active:           829104 kB
Inactive:         926416 kB
Active(anon):     789972 kB
Inactive(anon):   857132 kB
Active(file):      39132 kB
Inactive(file):    69284 kB
SwapTotal:       1052248 kB
SwapFree:         599844 kB
...
```

/proc (5)

Der Linux Kernel besteht aus einem Kern und dynamisch geladenen Modulen. Diese übernehmen Spezialaufgaben, zB Gerätetreiber.

```
$ cat /proc/modules
i915 67844 2 - Live 0xf81f6000
drm 96424 3 i915, Live 0xf8142000
nfs 266600 2 - Live 0xf8428000
lockd 74284 1 nfs, Live 0xf8102000
nfs_acl 11136 1 nfs, Live 0xf7d08000
sunrpc 195552 12 nfs,lockd,nfs_acl, Live 0xf8002000
input_polldev 11912 0 - Live 0xf7d4f000
lp 17156 0 - Live 0xf7d10000
usbkbd 13696 0 - Live 0xf7ce2000
snd_hda_intel 436148 2 - Live 0xf8320000
snd_pcm_oss 46336 0 - Live 0xf8281000
snd_mixer_oss 22656 1 snd_pcm_oss, Live 0xf8265000
...
```

/proc (6)

- `/proc/mounts` ... listet eingebundene Dateisysteme auf, so wie mount. Siehe auch `/etc/mtab`.
- Geräte:
  - `/proc/devices`
  - `/proc/bus/input/devices`
  - `/proc/bus/pci/devices` - besser lspci -v
  - `/proc/bus/usb/...` - besser lsusb -v
- `/proc/stat` ... Statistiken:
  - CPU: total, und für jede der CPUs, 1/100 Sekunden in Modus: user mode, user mode low priority, system mode, idle, iowait, ...
- `/proc/swaps` ... swap file Statistik
- `/proc/version` ... Kernel Version

/proc (7)

Jeder Prozess hat außerdem ein eigenes Unterverzeichnis (PID).
Darin befinden sich Informationen über ihn:

`/proc/<PID>/`

- cmdline: Aufruf
- cwd: current working directory
- environ: Umgebungsvariablen (export X="..")
- stat + status: running/idle, ...
- statm: Speicherverbrauch
- ...

/proc (8)

In /proc/net/ ist alles über das Netzwerk zu finden:

- dev: Netzwerk Devices
- arp: ARP Daten
- netstat: Netzwerk Statistik (gibt auch netstat Befehl!)
- sockstat: Socket Statistiken
- route: "Rohdaten" für den route Befehl
- udp/tcp: Sockets, ...
  - Besser netstat -udp bzw. netstat -tcp
- wireless: drahtlose Netzwerke
- ...

/proc (9)

In /proc/sys/ ist es möglich Werte zu ändern (mit `echo > ...`).

- Im Unterschied zu sysctl ist das der "direkte" Weg.

/proc/sys/net/X/Y ist äquivalent zum Key net.X.Y in sysctl.

/sys (sysfs)

Ab Kernel 2.6 gibt es das **sysfs**. Das ist eine Weiterentwicklung und wird auch dafür verwendet, um Applikationen Zugriff auf Geräte und deren Konfiguration zu geben. Es wird mittels mount eingebunden:

```
mount -t sysfs sysfs /sys
```

```
Layout:
/sys/
|-- block      ... Blockgeräte
|-- bus        ... PCI, ...
|-- class      ... graphics, input, ...
|-- devices    ... alle gefundenen Gerte
|-- firmware
|-- module     ... geladene Kernelmodule
‘-- power      ... Energiesteuerung
```

- Kernel sysfs Dokumentation
- The sysfs Filesystem by Patrick Mochel

RPC

RPC: "remote procedure call"

Erinnern wir uns zurück an das Kompilieren von C Programmen mit Bibliotheken. Solch eine Softwarebibliothek stellt Funktionen bereit, die von einem Programm abgerufen werden können - das ist ein "*procedure call*".
"Remote" wird es dann, wenn Bibliothek und das laufende Programm nicht mehr am selben Rechner läuft, sondern zB über das Netzwerk erfolgt.
Damit das funktioniert, muss die Applikation wissen, an welchem Port des anderen Rechners eine Server-Applikation läuft und auf Anfragen horcht.
RPC ist eine Spezielle Unix Technologie, jedoch gibt es viele analoge Systeme für diverse Programmiersprachen und Plattformen.

RPC /2

RPC: "remote procedure call"

- Client/Server müssen sich auf ein gemeinsames Protokoll für die "Serialisierung" der Daten und Befehle einigen. [Serialisierung ist das Kodieren von Datenstrukturen in einen einheitlichen Block von Bytes]. Das wird häufig über eine IDL (interface description language) gemacht.
- Unter Linux/Unix kann ein sogenannter "Portmapper" laufen, der anderen mitteilen kann an welchem Port welche Applikation läuft. Wichtig: Zuerst muss immer der Portmapper gestartet werden, dann der jeweilige Server der sich beim Portmapper registriert.

Wird verwendet für NFS (network file system), NIS/LDAP (Informationen über Benutzer, Passwörter, ...),

RPC /3

Portmapper:

```
$ rpcinfo -p program vers proto port
             100000     2   tcp  111 portmapper
             100000     2   udp  111 portmapper
             100003     2   udp 2049 nfs
             100003     3   udp 2049 nfs
             100003     4   udp 2049 nfs
  ...
```

Copyright & Lizenz