



Grundlagen der Linux-Systemadministration

Informatica Feminale, Bremen 2007

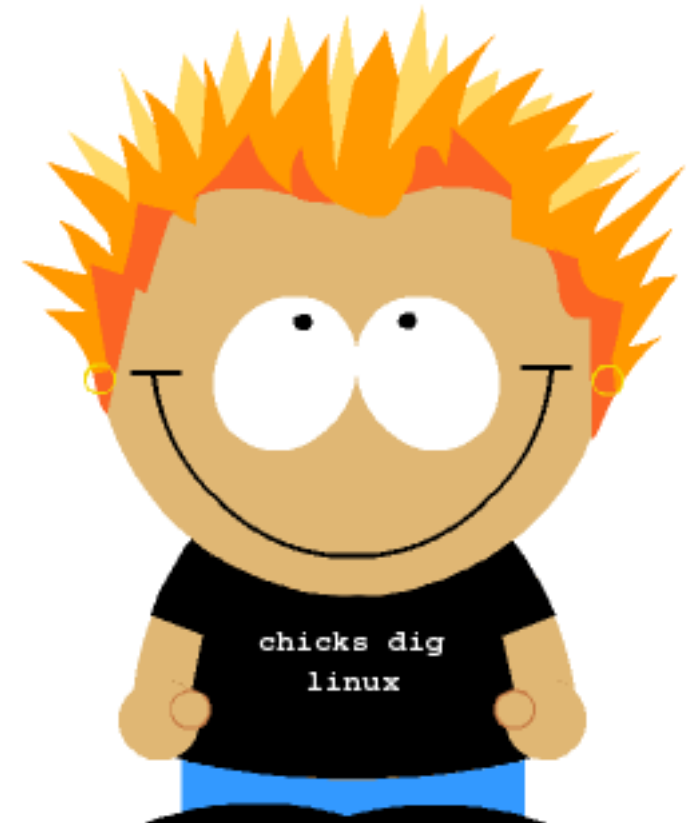
Jutta Horstmann

data in transit

I T - B E R A T U N G

Jutta Horstmann (Dipl. Inform., Dipl. Pol.)

- IT-Beraterin
- Schwerpunkt: Freie Software
- zuerst: EDV-Support
- dann: Systemadministration,
Datenbankadministration
- dann: Datenbank-Entwicklung
- dann: selbständig



data in transit

- gegründet 2005 in Berlin
- 2006 Umzug nach Bonn

data in transit
I T - B E R A T U N G

- Schwerpunkte:

Web-Programmierung, CMS, Datenbanken

Konzeption, Systemarchitektur, Requirement Engineering

- Kunden:

synthax Audio, relevantive, apikula, fczb, MSH AND MORE, papaya Software, hexabinær, dbpr, Zoologisches Forschungsmuseum Alexander Koenig, Wissenschaftszentrum Berlin, tarent, intevation, DesignWork, aurlSP ...

- Referenzen:

<http://www.dataintransit.com/de/projects>



Die Themen




Systemadministration – Was ist das?



Als Adminine einloggen



Benutzerverwaltung
Gruppenverwaltung
Rechteverwaltung



Prozesse & Prozess-Verwaltung



Systemprozesse (Dämonen)



Prozesse automatisch starten: Cronjobs



Logging & Monitoring



Backup & Recovery



Weitere zentrale Aufgaben



Systemadministration – Was ist das?

Definition (allgemein)

Systemadministratorinnen

- planen,
- installieren,
- konfigurieren und
- pflegen
- die informationstechnische Infrastruktur eines Unternehmens oder anderer Organisationen.

(Quelle: Wikipedia)

Aufgaben der Admine

Aufgaben sind unter anderem

- Benutzerbetreuung und Benutzerverwaltung,
- Management, Erweiterung und Erneuerung von Hard- und Software,
- Backup und Recovery,
- Integration von Systemen ins Netz,
- Leistungsmessung und
- Anpassung je nach Einsatzzweck



Als Adminine einloggen

Shell, su, sudo

- Shell
 - lokaler Rechner: Kommandozeile öffnen
 - entfernter Rechner: `ssh <username>@<hostname>`
- `su` – (dasselbe wie `su -l`)
- `sudo` + Befehl (z.B. `sudo shutdown -r now`)
 - Login ohne root-Passwort
 - Logging
 - feingranular root-Rechte vergeben
 - `visudo` (/etc/sudoers editieren): `root ALL=(ALL) ALL`

Aufgabe: sudo

1. Logge Dich als root ein
2. Öffne `/etc/sudoers` zum editieren
3. Erweitere die Datei um Deinen Benutzernamen und weise Dir alle root-Rechte zu
4. Beende Deine root-Sitzung
5. Öffne `/etc/sudoers` zum Editieren mittels des “sudo”-Befehls
6. Beende das Editieren

Lösung: sudo

1. `su -`
2. `visudo`
3. `i; <username> ALL=(ALL) ALL; esc;`
`:wq`
4. `exit`
5. `sudo visudo`
6. `:q`



Benutzerverwaltung Gruppenverwaltung Rechtemanagement

Linux: Ein Mehrbenutzersystem

- Arbeitsumgebungen für verschiedene Benutzer
- voneinander abgegrenzt
- eingeschränkte Rechte
- Voreinstellungen für Benutzer

Alles ist eine Datei

- Zugang zu Dateien wird über Dateirechte gesteuert
- Zugang zu Programmen ebenfalls
- Dateirechte sagen aus
 - welche Nutzerin
 - welche Gruppe
 - was mit der Datei machen darf: Lesen, Schreiben, Ausführen

Aufgaben: Umgucken

1. Wer bist Du? Wo bist Du? (Nutzername, Home-Verz.)
2. Was ist Deine UID, GID?
3. Was ist Deine Shell?
4. Wie viele NutzerInnen gibt es sonst noch auf dem System? Welche?
5. Wer ist aktuell noch eingeloggt? Was machen sie?
6. Was für Gruppen gibt es auf dem System? Welche GID, welche Gruppenmitglieder?
7. Zu welchen Gruppen gehörst Du?

Lösung: Umgucken

1.whoami; id; cd; pwd

2.getent passwd <username>

3.getent passwd <username>

4.getent passwd | wc -l;
cat /etc/passwd | cut -d: -f1,5 | tr
": " "\t"

5.w; who

6.getent group

7.groups

Aufgabe: Nutzerin/Gruppe verwalten (1)

1. Lege eine neue Nutzerin an:

- Name: Anna B. Blume
- username: annab
- Passwort: Denk Dir eins aus
- Home-Verzeichnis /home/annab
 - Welche Daten werden automatisch im Home angelegt?
 - Wo kann man das konfigurieren?

2. Ändere den Namen in Anna Berta Blume

3. Ändere den Nutzernamen in annabb

Aufgabe: Nutzerin/Gruppe verwalten (2)

1. Lege eine neue Gruppe an:
 - Name: informatica
2. Weise Anna dieser Gruppe zu
3. Entferne Anna aus der Gruppe
4. lösche die Gruppe
5. lösche Anna inkl. ihres Home-Verzeichnisses

Lösung: Nutzerin/Gruppe verwalten (1)

1. `sudo useradd -d /home/annab -m annab`
 - `ls -la /home/annab`
 - `more /etc/default/useradd`
 - `ls /etc/skel`
2. `sudo chfn -f "Anna Berta Blume" annab`
3. `sudo usermod -l annabb annab; getent passwd annab; getent passwd annabb`

Lösung: Nutzerin/Gruppe verwalten (2)


1. `sudo groupadd informatica; getent group informatica`
2. `sudo groupmod -A annabb informatica`
 - Checken: `groups annabb`
3. `sudo groupmod -R annabb informatica`
4. `sudo groupdel informatica`
5. `sudo userdel -r annabb`

Dateirechte

- ls -la: Zeigt auch Besitzerin, Gruppe und Rechte an
- Bsp.: `-rwxr--r--`
 - kein Verzeichnis, sondern Datei
 - Besitzerin hat Lese-/Schreib- und Ausführrechte
 - Gruppe und Alle haben Leserechte
- Bsp.: `drwx-----`
 - Verzeichnis
 - nur Besitzerin hat Rechte

Rechte und Besitzerin ändern mit chmod, chown, chgrp

- `chmod u+w` (User bekommt Write-Rechte)
- `chmod g-r` (Gruppe bekommt Leserechte entzogen)
- `chmod a+x` (Alle bekommen Ausführ-Rechte)
- Alternativ mit Zahlen:
 - 7: lesen, schreiben, ausführen (111 binär = 7)
 - 6: lesen, schreiben (110 binär = 6)
 - 4: lesen (100 binär = 4)
 - **Was bedeutet:** 755? 666? 600? 640?
- `chown <username>`, `chgrp <gruppenname>`



Prozesse & Prozess-Verwaltung

Was ist ein Prozeß?

- Ablauf eines Programms im Computer
- benötigt
 - Speicherabbild des Programms
 - Speicher für die Daten
 - vom Betriebssystem bereitgestellte Ressourcen
 - Prozessor
- Zustände: dead, ready, running, sleep, trace, wait, uninterruptible sleep, zombie
- Ende eines Prozesses: natürlich, durch Nutzer, durch Fehler, aus System-Gründen

Beschreibung eines Prozesses

- PID (Prozess-ID)
- PPID (Parent PID)
- UID, EUID: reale und effektive Benutzer-ID
- GID, EGID: reale und effektive Gruppen-ID
 - real: wer den Prozess erzeugt hat
 - effektiv: unter welchen Rechten der Prozess gerade läuft
- nice-Wert (Verhältnis zur Priorität anderer Prozesse)

Prozeßüberwachung mit ps

- Standardaufruf: `ps aux`
- `ax`: alle Prozesse, `u`: user-friendly output

```
USER      PID  %CPU  %MEM  VSZ   RSS TTY  STAT  START   TIME  COMMAND
root        1   0.0   0.0   588    56 ?    S     2006   37:10  init [3]
root        2   0.0   0.0     0     0 ?    SN    2006    0:01  [ksoftirqd/0]
root        3   0.0   0.0     0     0 ?    S<    2006  335:46  [events/0]
root        4   0.0   0.0     0     0 ?    S<    2006    0:00  [khelper]
root        5   0.0   0.0     0     0 ?    S<    2006    0:00  [kacpid]
root     4337   0.0   0.4  4436  1124 ?    S     2006    0:11  /usr/sbin/saslauthd -a pam
root     6159   0.0   0.1 46624   336 ?    Ss    2006  109:55  /usr/sbin/httpd2-prefork -f
                                     /etc/apache2/httpd.conf
root     7137   0.0   0.2  4176   584 ?    Ss    Jun21    6:55  /usr/lib/postfix/master
postfix 7149   0.0   0.3  4360   752 ?    S     Jun21    3:34  qmgr -l -t fifo -u
jh     11671   0.2   2.1  7088  5336 ?    Ss    20:55    0:07  imapd
wwwrun 11781   0.1   2.8 49260  6996 ?    S     21:25    0:01  /usr/sbin/httpd2-prefork -f
                                     /etc/apache2/httpd.conf
```

- USER: Benutzername der Prozeßeigentümerin
- PID: Prozeß-ID
- % CPU-Belastung
- % des durch Prozeß belegten Speichers
- VSZ: Virtuelle Prozeßgröße (KB)
- RSS: Größe der residenten Menge (Anzahl 1KB-Seiten im Speicher)
- TTY: ID des Steuerterminals
- STAT: aktueller Prozess-Status

Prozeßüberwachung mit top

```
top - 22:04:25 up 2:13, 7 users, load average: 2.97, 2.45, 1.64
Tasks: 97 total, 4 running, 93 sleeping, 0 stopped, 0 zombie
Cpu(s): 47.7%us, 4.5%sy, 0.0%ni, 22.7%id, 11.4%wa, 4.5%hi, 9.1%si, 0.0%st
Mem: 499772k total, 490244k used, 9528k free, 13068k buffers
Swap: 1052216k total, 33748k used, 1018468k free, 151152k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
5480	jh	15	0	140m	53m	24m	S	5.0	11.0	2:48.53	amarokapp
4973	jh	16	0	247m	97m	20m	R	1.0	20.0	3:50.20	firefox-bin
3771	root	16	0	315m	43m	4312	R	0.7	8.9	1:36.12	Xorg
4929	jh	15	0	34144	17m	12m	R	0.7	3.5	0:03.54	konsole

Prozesse beenden

- `kill -<Signal> PID`
 - höflich: `kill -15` (TERM, Standardwert).
Programm soll sich selbst beenden
 - bestimmt: `kill -9` (KILL). Betriebssystem soll
Programm beenden.
 - `kill -1` (HUP). Programm soll sich selbst resetten
(beenden und neu starten)
- `killall -<Signal> Prozeßname`: Beendet
alle Prozesse dieses Namens

Nützliche Helferlein in der Prozeßverwaltung

- `pidof <Kommando>`, z.B. `pidof mysqld`
- `nice`, `renice`:
 - hoher Wert: Prozeß ist sehr freundlich zu anderen Prozessen bezüglich seiner CPU-Zeit
 - heutzutage meist nicht mehr relevant
 - Werte von -20 bis +20, Standard: 0

Aufgabe: Prozesse

1. Starte einen Browser-Prozess
2. Gehe auf
<http://www.youtube.com/watch?v=LAr3XbqUbjo>
(Variante ohne GUI: `yes hallo`)
3. Gib Informationen über den Browser-Prozess aus
4. Betrachte die durch den Browser-(/yes-)Prozess verursachte Systemauslastung
5. Beende den Browser-Prozess (`yes`-Prozess)
6. Betrachte die Veränderung in der Systemauslastung

Lösung: Prozesse

1. `firefox&`
2. `firefox`
`http://www.youtube.com/watch?v=LAr3XbqUbj0 &`
3. `ps aux | grep firefox`
4. `top`
5. `pidof firefox; kill PID` oder `killall firefox`
oder "K" + PID in top
6. `top`



Systemprozesse (Dämonen)

Was sind Dämonen (daemons)?

- Programme, die im Hintergrund laufen und Dienste zur Verfügung stellen
- starten durch
 - Systemstart
 - von Hand
 - per inetd / xinetd: Dämon, der andere Dämonen startet,
- Serverdienste (z.B. cupsd, dhcpd, httpd, named, portmap, sshd, squid, vsftpd...)
- Kernel-Prozesse (z.B. aio, events, kapmd, kjournald, kswapd, pdflush)

init: Der “Ur-Prozeß”

- erster Prozeß, der bei Boot gestartet wird (PID 1)
- definiert “Run Levels”: welche Systemressourcen werden aktiviert
- Konfiguration in `/etc/inittab` und `/etc/init.d/*` bzw. `/etc/rcX.d/*`
- ansprechen: z.B.
 - `init 1`: Single-user mode
 - `init 3`: Full multiuser with network (keine grafische Oberfläche) - **Ausprobieren!**
 - `init 5`: Full multiuser with network and xdm (mit GUI)

Dämonen beim Booten starten

- entweder: Einrichten von Link zum Startskript des Daemons innerhalb von `/etc/rc.d`
- einfacher: `chkconfig`
- Aufgaben:
 - lass Dir alle beim Booten startenden Daemonen anzeigen (`chkconfig --list`)
 - entferne den Drucker-Spooler (`sudo chkconfig --del cups`)
 - füge den Drucker-Spooler wieder hinzu (`sudo chkconfig --add cups`)

Internet-Dämonen: xinetd

- früher: inetd
- jetzt: Extended Internet Service Daemon
- 2 Arten von Internet-Diensten
 - laufen permanent im Hintergrund, überwachen IP-Ports (Bsp: httpd 80, sshd 21...)
 - starten nur bei Bedarf – gesteuert durch xinetd (z.B. vsftpd, smtpd)

Aufgabe: Daemon neu starten

1. überprüfe, ob der Drucker-Spooler (cupsd) auf Deinem System läuft
2. Beende den Druckerdaemon
3. Starte den Druckerdaemon
4. Überprüfe den Prozess (sein Laufen und seine Nutzung der System-Ressourcen)

Lösung: Daemon neu starten

1. `ps aux | grep cups`
2. `sudo /etc/init.d/cupsd stop` oder (SuSE)
`sudo rccupsd stop` oder (Red Hat, Fedora,
Mandriva) `sudo service cupsd stop`
3. wie oben mit “start” (bzw. statt “stop” und “start”
einfach “restart”)
4. `ps aux | grep cups; top`
 - Mögliche Parameter: start, stop, restart, reload, status
 - reload: Einlesen von Config-Datei *ohne* Daemon-Stop



Prozesse automatisch starten: Cronjobs

Was ist cron?

- ein Dämon, beim Systemboot gestartet
- liest jede Minute Steuerungsdateien aus
- startet bei Bedarf darin angegebene Prozesse
- globale Konfiguration in `/etc/crontab`
- Benutzerspezifische crons (auch root): `/var/spool/cron/tabs`
- dort nicht editieren!

Die Cron Tabelle editieren

- root: `sudo crontab -e`
- Nutzer-Crons:
 - als Nutzer eingeloggt: `crontab -e`
 - oder `sudo crontab -u <username> -e`
- nur lesen: `crontab -l`
- crontab löschen: `crontab -r`

Aufbau der crontab

```
SHELL=/bin/sh
PATH=/usr/bin:/usr/sbin:/sbin:/bin:/usr/lib/news/bin
MAILTO=root
-*/15 * * * *    root    test -x /usr/lib/cron/run-crons &&
                  /usr/lib/cron/run-crons >/dev/null 2>&1
59 * * * *      root    rm -f /var/spool/cron/lastrun/cron.hourly
14 4 * * *      root    rm -f /var/spool/cron/lastrun/cron.daily
29 4 * * 6      root    rm -f /var/spool/cron/lastrun/cron.weekly
44 4 1 * *      root    rm -f /var/spool/cron/lastrun/cron.monthly
| | | | |      |    |
min h  d m wd   user  command
```

- **Minute, Stunde, Tag, Wochentag (0 = Sonntag) der Ausführung. * = jeder**
- **Nutzer, mit dessen Rechten der Prozeß läuft**
- **Kommando zum Starten des Prozesses**

Aufgabe: cron

- sind auf Deinem Rechner Cronjobs konfiguriert?
 - vom wem?
 - wann laufen sie?
 - was tun sie?
- richte Dir einen Cronjob ein, der
 - immer Di, Mi und Do, in jeder geraden Stunde zwischen 8h und 18h zur 5., 30, und 40. Minute ...
 - eine Mail mit den letzten Zeilen des Syslogs ...
 - an root (bzw. per Alias an Dich) versendet

Lösung: cron

- (sudo) crontab -l
sudo ls /etc/cron.X/
sudo more /var/spool/cron/tabs/X
- sudo crontab -e
5,30,40 8-18/2 * * 2-5 tail /var/log/messages
- Exkurs: Mail-Alias
 - sudo vi /etc/aliases
 - root: <username_oder_mailadresse>, \root



Logging & Monitoring

Was ist das?

Wozu braucht man das?

- Logging/Protokollieren:
 - Eine Logdatei beinhaltet das **automatisch erstellte Protokoll** aller oder bestimmter Aktionen von Prozessen auf einem Computersystem.
- Monitoring/Überwachen:
 - bei einem **beobachteten Ablauf bzw. Prozess steuernd einzugreifen**, sofern dieser nicht den gewünschten Verlauf nimmt bzw. bestimmte Schwellwerte unter- bzw. überschritten sind

Die wichtigsten Log-Dateien

- Hauptfundort: `/var/log/`
 - `boot.log`, `faillog`, `localmessages`, `mail`, `messages`, `vsftpd.log`, `warn`, `xinetd.log` ...
 - `/var/log/apache2/access_log` und `error_log`
- Sonstige:
 - `/var/lib/mysql/mysqld.log`, `/opt/Zope/log/event.log`, `/usr/lib/rpm/rpm.log`, ...

Logging mit syslog

- System und Anwendungen nutzen syslog-Bibliothek und -Daemon
- Vorteil: zentrale Steuerung des Loggings
- syslogd
- /etc/syslog.conf bzw. /etc/syslog-ng/syslog-ng.conf
- was soll wohin geloggt werden
- setzt Zeitstempel (- MARK -) in Logs
- Prioritäten (emerg, alert, crit, err, warning, notice, info, debug)

/var/log/messages

```
Sep  1 12:59:00 zoe /USR/SBIN/CRON[23175]: (root) CMD ( rm -f
/var/spool/cron/lastrun/cron.hourly)
Sep  1 13:13:07 zoe PAM-warn[4336]: function=[pam_sm_authenticate]
service=[smtp] terminal=[<unknown>] user=[jh] ruser=[<unknown>]
rhost=[<unknown>]
Sep  1 13:26:54 zoe -- MARK --
Sep  1 13:46:41 zoe xinetd[23364]: warning: /etc/hosts.allow, line 64:
can't verify hostname: getaddrinfo
(unknown76.120.65.69.defenderhosting.com): Name or service not known
Sep  1 13:46:41 zoe xinetd[23364]: libwrap refused connection to ftp
(libwrap=vsftpd) from 69.65.120.76
Sep  1 13:59:00 zoe /USR/SBIN/CRON[23392]: (root) CMD ( rm -f
/var/spool/cron/lastrun/cron.hourly)
Sep  1 14:07:08 zoe sshd[23423]: Accepted keyboard-interactive/pam for
jh from 89.52.160.221 port 61792 ssh2
Sep  1 14:09:11 zoe sudo:          jh : TTY=pts/0 ; PWD=/home/jh ;
USER=root ; COMMAND=/usr/bin/tail -50 /var/log/messages
```

Was tun mit den Dateien?

- wegschmeissen
- nach einer bestimmten Zeit wegschmeissen
- nach einer bestimmten Zeit komprimieren
- nach einer bestimmten Zeit komprimieren und auf einem externen Datenträger archivieren

Praktisch: **Rotation**

- Logs dieser Woche in aktuell.log, letzte Woche alt1.log, vorletzte alt2.log, vorvorletzte alt3.log
- Zum Wochenende wird alt3.log gelöscht und die anderen werden umbenannt

Nützliche Helferlein

- Lesen von Logs:
 - `tail`
 - `head`
- Lesen von `last.log`: `last`
- Logs verwalten: `logrotate`

Aufgabe: Logging

1. Lass Dir die letzten 100 Zeilen der messages anzeigen und erkläre, was dort passiert.
2. Lass Dir die ersten 100 Zeilen des mail-Logs anzeigen und erkläre, was dort passiert.
3. Ist auf Deinem Server `logrotate` im Einsatz und wenn ja, mit welcher Strategie?
4. Zeige die letzten 10 Logins an. Wer? Wann? Woher? Wie lange?

Lösung: Logging

1. `sudo tail -100 /var/log/messages | more`
2. `sudo head -100 /var/log/mail | more`
3.
 - `sudo which logrotate;`
 - `ls /etc/cron.daily;`
 - `more /etc/cron.daily/logrotate;`
 - `more /etc/logrotate.conf;`
 - `ls /etc/logrotate.d/`
4. `last -10`

Monitoring: Was wollen wir überwachen?

- Log-Ins
- Systemzustand:
 - Hardware, z.B. Temperatur
 - Speicher, z.B. Plattenplatz
 - Auslastung, z.B. CPU, RAM, Netzwerk-Traffic
- Sicherheit
 - unberechtigte Zugriffe

Monitoring-Strategien und -Tools

- Logs
 - regelmäßiges Log-Lesen
 - Senden von Log-Meldungen (Kommandozeile, Mail...)
 - Log-Auswertungs-Tools: `swatch`, `logcheck`...
- Plattenverbrauch: `df -k`, `du -s`
- Prozesse: `ps`, `top`
- Speicher: `free` (`watch -n 1 -d free`), `vmstat`
- System: `sysstat`, `uptime`
- Netzwerk: `ping`, `traceroute`, `tcpdump`, `ntop`

Monitoring-Tools: Links und Listen

- Allgemein:
 - <http://www.debianhelp.co.uk/monitortools.htm>
 - <http://www.informit.com/articles/article.aspx?p=29666&seqNum=8>
 - Munin (<http://munin.projects.linpro.no/>)
- Systemressourcen:
 - <http://www.novell.com/coolsolutions/tools/downloads/linux-monitoring.tar.bz2>
 - http://www.volny.cz/linux_monitor/
- GUI-Tools:
 - gnome-system-monitor
 - KDE-Systemüberwachung (KsysGuard)
 - xload, xosview
- Netzwerk:
 - <http://www.slac.stanford.edu/xorg/nmtf/nmtf-tools.html>
 - <http://www.ubuntugeek.com/bandwidth-monitoring-tools-for-linux.html>
 - <http://www.topology.org/comms/netmon.html>



Backup & Recovery

Backup-Strategien

- Was soll gesichert werden?
Alles? /home ja, /tmp nein ...? Nur die Änderungen (“inkrementell”)?
- Wie oft soll es gesichert werden? *stündlich? täglich?*
- Wann soll es gesichert werden? *nachts?*
- Wohin soll es gesichert werden? *anderer Rechner? DVD? Band?*
- Wo und wie wird die Sicherung aufbewahrt?
selbes/anderes Gebäude? einbruchssicher? feuerfest?
- Wie lange soll es dort gesichert bleiben?
eine Woche? ein Jahr? 10 Jahre? Immer?
- Funktioniert die Sicherung??!! *(Methode, Zustand der Bänder)*

Einfache Backups mit tar oder cpio

- Sichern:

- Archiv erstellen:

- `tar -cf <name des tar-archivs> /home/jh`

- Komprimieren: `gzip`, `bzip2`, `zip`

- auf externen Datenträger kopieren (`cp`, `ftp`, `sftp`, `rsync`)

- alles zusammen: `tar -czf /dev/tape /home/jh`

- Wieder herstellen:

- `gunzip`, `bunzip2`, `unzip`, `tar -xzf`, `tar -xjf`

- Alternative: `cpio`

- erzeugt Liste aller Dateien, die seit letzter Sicherung verändert wurden
 - packt diese in einzige grosse Datei
 - archiviert diese auf einem externen Gerät
 - sichert Dateien jeden Typs
 - behält Berechtigungen, Eigentum und Änderungszeiten bei
 - Backup-Level: Wert zwischen 0 und 9.
 - z.B. Level-5-Backup erzeugt Backup für alle Dateien, die sich seit der letzten Sicherung eines Levels <5 geändert haben.
 - Speichert Datum, Level und Dateisystem in /etc/dumpdates

- einzelne Dateien wiederherstellen
 - zugehöriges Band ausfindig machen
 - temporäres Verzeichnis `/var/restore`
 - `restore i` (interaktiv)
 - Dateien auswählen mit `add`, Dateien abrufen mit `extract`
- Dateisystem wiederherstellen
 - Beginnen mit aktuellster Level-0-Sicherung
 - `restore r`
 - danach: inkrementelle Sicherungen in der Reihenfolge ihrer Erstellung

Beispiel: Backup mit Redundanzen

- Erster Montag des Monats: Level 0 (Vollbackup)
- Jeder andere Montag: Level 1
 - (wöchentliches inkrementelles Backup relativ zu Level 0)
- Dienstag: Level 2
 - (tägliches inkr. Backup relativ zu Level 1 – alles seit Montag)
- Mittwoch: Level 2
 - (tägliches inkr, Backup relativ zu Level 1 – alles seit Montag)
- Donnerstag: Level 2
 - (tägliches inkr. Backup relativ zu Level 1 – alles seit Montag)
- Freitag: Level 2 (tägl. inkr. Backup rel. zu Level 1 – alles seit Montag)

Beispiel: Crash am Donnerstag

- Wiederherstellen
 - des letzten Vollbackups (Level 0)
 - des letzten Wochenbackups (Level 1)
 - des letzten Tagsbackups (Level 2, Mittwoch)

Aufgabe: BOFH

“It's backup day today so I'm pissed off. Being the BOFH, however, does have it's advantages. I reassign null to be the tape device - it's so much more economical on my time as I don't have to keep getting up to change tapes every 5 minutes. And it speeds up backups too, so it can't be all bad can it? Of course not.” (<http://bofh.ntk.net/Bastard1.html>)

- Welcher Befehl wurde hier ausgeführt?
- Was passiert dabei?
- Wie geht es richtig?



Weitere zentrale Aufgaben

- Netzwerk
- Sicherheit
- verschiedene Server-Software:
 - Mail
 - Webserver
 - Datenbanken
- Konfigurationen für die User

Literatur

- Ellen Siever et al.: Linux in a Nutshell. O'Reilly
- Evi Nemeth et al.: Handbuch zur UNIX Systemverwaltung. Prentice Hall / Markt & Technik
- Eelen Frisch: Essential System Administration. O'Reilly
- Michael Kofler: Linux. Addison-Wesley
- <http://www.dataintransit.com/de/linux-training-links>

Fragen???



data in transit <http://www.dataintransit.com>
Kontakt jh@dataintransit.com