UNIX - Systemadministration am Beispiel von OSF/1

Helmuth Blasch blasch@zid.tu-graz.ac.at

v1.1, 07 Feb 1998

Dieser Kurs soll eine Einführung Anhand von Digital Unix (ehm. OSF/1) in den Aufgabenbereich eines Systemadministrators geben.

Contents

1 UNIX-Geschichte

1.1 Ursprung

Die Entwicklung von UNIX begann bereits 1969 in der Entwicklungsabteilung der amerikanischen Telefongesellschaft AT&T. Es sollte ein modernes Betriebssystem werden, das über Multiuser- und Multitasking-Eigenschaften verfügt.

Die Verbreitung des Betriebssystems wurde durch die Portierbarkeit, die kostengünstige Lizensierung und grosszügige Weitergabe an Universitäten unterstützt. Bereits 1973 bestand der UNIX-Quellcode zu etwa 90% aus der Programmiersparche **C** und nur noch wenige systemnahe Teile aus Assemblercode, was die Portierbarkeit auf andere Hardware Platformen stark erleichterte.

1.2 Entwicklung

Neben der Entwicklung bei AT&T, die heute als System V bezeichnet wird, begann ab 1975 an der University of California, Berkeley die Entwicklung des Systems BSD. Dabei wurde auf Netzfunktionalität und ein neues Filesystem Rücksicht gelegt.

Neuere UNIX Derivate besitzen Bestandteile von beiden Entwicklungsrichtungen und können nicht eindeutig zugeordnet werden. Auch OSF/1 hat von beiden Richtungen Teile übernommen, wobei die BSD Seite überwiegt.

System V

- SunOS
- Solaris
- HP-UX
- IRIX
- AIX
- Linux

BSD

- Ultrix
- OSF/1 (Digital Unix)

1.3 Eigenschaften

- Der Kernel: er wird beim Booten permanent in den Hauptspeicher geladen und verwaltet den Rechner und dessen Peripherie. Er kontrolliert die Prozesse und implementiert die Multiuser- und Multitasking-Eigenschaften des Systems. Dabei wird Multitasking über die Verwendung von Time-Slices zu Verfügung gestellt. Bei Symetrical Multiprozessor Machines (SMP) wird zusätzlich eine Verteilung auf die vorhandenen Prozessoren (echtes Multitasking) vorgenommen. Der Kernel stellt ausserdem Device-Files zur Verfügung, die den Zugriff auf die Peripherie erlauben (sowohl von der `Shell` als auch aus einem `C-Programm`).
- Über dem Kernel liegt ein grosser Satz an Hilfsprogrammen, die dem Benutzer die Interaktion mit dem System erlauben. Das wichtigste dabei ist die Shell. Sie erlaubt auch komplexere Operationen ohne ein Hilfsprogramm in einer höheren Sprache schreiben zu müssen.

- Das Filesystem erlaubt schnellen und einfachen Zugriff auf Daten und gestattet es ausserdem den Zugriff auf die Daten zu kontrollieren.
- Die Ein- und Ausgabe ist - relativ - Device-unabhängig. Es macht keinen Unterschied ob man auf ein File oder ein Band schreibt.
- Weiters erlaubt UNIX die Interprozesskommunikation, stellt Netzanbindung und graphische Oberfläche zur Verfügung.

2 Start- und Shutdown

2.1 Booten (Start)

Nach dem Einschalten der Maschine beginnt diese je nach Konfiguration (siehe 2.5) automatisch oder nach der Eingabe des Kommandos `boot`, zu booten. Das Booten der Maschine läuft in mehreren Schritten ab, die in den nachfolgenden Paragraphen beschrieben sind.

2.1.1 Laden des Kernels

Zunächst wird der Kernel von der Platte in den Hauptspeicher geladen. Welcher Kernel, von welcher Platte geladen wird kann über die bootconsole (siehe 2.5) festgelegt werden. Das kann bei einem defekten Kernel helfen einen alternativen Kernel zu laden oder bei einer defekten Platte von einer alternativen Platte zu booten.

Das Laden des Kernels läuft in mehreren Schritten ab. Zuerst wird nur ein Boot Programm `osf-boot` geladen - *primary boot*. In ihm befinden sich die Startroutinen die den eigentlichen Systemkern `vmunix` von der Platte in den Hauptspeicher übertragen und starten - *secondary boot*.

2.1.2 Prüfen der Hardware

Nach dem Laden des Kernels beginnt dieser mit dem Analysieren und Testen der vorhandenen Hardware. Dabei wird vor allem der Hauptspeicher und die Festplatten überprüft sowie die Grösse des Hauptspeichers bestimmt und eine Reservierung der I/O Buffer vorgenommen. Wichtig dabei ist, dass bereits alle Devices in Betrieb sind, da zu diesem Zeitpunkt der Kernel die Device Treiber der vorhandenen Geräte aktiviert. Ein späteres Einschalten eines Devices oder das Dazuhängen eines Gerätes während dem Betrieb ist daher nicht möglich. Sehr wohl ist es möglich ein Gerät (z.B.: Tapedrive) während des Betriebs auszuschalten sofern es nicht ein Swapdevice ist.

Das Testen der Festplatten beschränkt sich dabei auf deren Ansprechbarkeit und nicht auf deren Oberfläche noch auf deren Filesystem. Dies wird nur durch `newfs` (Oberfläche) bzw. in den Startup Files durch `fsck` (Filesystem siehe 2.2.2) veranlasst.

2.1.3 Starten der Prozesse

Sind die Checks durchgeführt wird der Kernel Prozess gestartet. Er bekommt die PID 0 (**P**rocess**ID**entification). Danach wird das Root Filesystem gemountet und das Programm `init` (siehe 2.2) gestartet. Es bekommt die PID 1.

2.2 init

Der Prozess `init` ist der Mutterprozess aller folgenden Prozesse. Er ist der letzte Schritt im Bootprozess und wird gleich nach dem Mounten des Root Filesystems ausgeführt. Der Prozess Init wertet das Konfigurationsfile `inittab` (siehe 2.2.1) aus. In diesem sucht es zuerst nach dem Eintrag `initdefault`. Dieser bestimmt den Runlevel (siehe auch 2.2.3) in welchem das System nach Abschluss der Bootphase laufen soll. In OSF/1 gibt es die Runlevels 0, 2, 3 und s. Der Runlevel 0 steht für Shutdown und Runlevel S bzw. s für den Single User Mode (siehe 2.4). Die Runlevel 2 und 3 sind Multiuser Level. Soll die Datei `inittab` während des Betriebes neu gelesen werden so kann der Befehl `init q` verwendet werden.

2.2.1 inittab

Die Datei `inittab` im Verzeichnis /etc besitzt die Syntax

```
ID:run_levels:action:process
```

- `ID` wird dazu benutzt dem Objekt bestehend aus run_levels, action und process einen Namen zu geben. Dieser kann 14 Zeichen lang sein.
- Das Feld `run_levels` wird dazu benützt die Runlevel aufzuzählen, die diese Zeile ausführen sollen. So bedeutet `..:23s:..`, dass die Zeile sowohl im Runlevel 2, 3 als auch im Single User Mode ausgeführt wird, beziehungsweise bei einem Wechsel in den jeweiligen Modus.
- Das Feld `action` legt die Art und Weise fest mit der der nachfolgende Befehl im Feld `process` abgearbeitet werden soll. Das Feld `action` kann dabei folgende Werte haben:

  respawn
  gibt an, dass der Befehl, falls er nicht exisiert oder stirbt, von `init` gestartet wird. Existiert der Prozess so fährt `init` mit der Abarbeitung von `inittab` fort.

  wait
  gibt an, dass `init` beim Wechsel des Runlevels den Befehl ausführt und auf dessen Beendigung wartet.

  once
  gibt an, dass `init` beim Wechsel des Runlevels den Prozess startet aber nicht auf dessen Beendigung wartet und ihn auch nicht 'restartet' wenn der Prozess stirbt. Läuft der Prozess beim Wechsel in einen anderen Runlevel bereits so wird er nicht nocheinmal ausgeführt.

  boot
  gibt an, dass `init` diese Zeile beim erstenmal lesen der Datei `inittab` ausführt und nicht auf die Beendigung des Prozesses wartet. Wenn der Prozess stoppt wird er nicht durch `init` 'restartet'. Der Runlevel muss der default Runlevel sein oder der Runlevel in dem sich das System beim booten befindet.

  bootwait
  wie bei `boot` jedoch wartet `init` auf die Beendigung des Prozess.

  powerfail
  die Zeile wird ausgeführt wenn `init` das Signal `SIGPWR` erhält.

  powerwait
  die Zeile wird ausgeführt wenn `init` das Signal `SIGPWR` erhält. `Init` wartet auf die Beendigung des Prozesses bevor es mit der Ausführung der Datei `inittab` fortfährt.

off

existiert ein Prozess der mit dem Prozess, welcher in dieser Zeile definiert ist, übereinstimmt so wird ihm zuerst ein Warnsignal `SIGTERM` gesandt. Nach 20 Seckunden sendet `init` ihm ein `SIGKILL`. Existiert der Prozess nicht wird die Zeile ignoriert.

initdefault

legt den default Runlevel fest. Ist kein Eintrag im Feld run_levels vorhanden so interpretiert `init` den Runlevel als 0s23. Es wird immer der höchste Eintrag als default Runlevel gewählt. Im letzten Fall also 3. Fehlt der `initdefault` Eintrag überhaupt so fragt `init` den Operator an der Konsole nach dem Runlevel.

sysinit

Einträge dieses Typs werden von `init` ausgeführt bevor `init` versucht die Konsole anzusprechen. Dieser Eintrag kann also dazu benutzt werden um Devices zu konfigurieren.

- # leitet ein Kommentar ein
- Im Feld `prozess` kann ein `shell` Kommando eingetragen werden. es wird die /bin/sh dazu verwendet es auszuführen.

Beispiel:

```
# @(#)$RCSfile: inittab,v $ $Revision: 4.1.17.2 $ (DEC) $Date: 1994/08/29 15:24:19 $
#
#
# (c) Copyright 1990, OPEN SOFTWARE FOUNDATION, INC.
# ALL RIGHTS RESERVED
#
#
# OSF/1 Release 1.0
is:3:initdefault:
ss:Ss:wait:/sbin/rc0 shutdown < /dev/console > /dev/console 2>&1
s0:0:wait:/sbin/rc0 off < /dev/console > /dev/console 2>&1
fs:23:wait:/sbin/bcheckrc < /dev/console > /dev/console 2>&1
kls:Ss:sysinit:/sbin/kloadsrv -f & < /dev/console > /dev/console 2>&1
sysconfig:Ss:sysinit:/sbin/init.d/autosysconfig start < /dev/console > /dev/console 2>&1
update:23:wait:/sbin/update > /dev/console 2>&1
it:23:wait:/sbin/it < /dev/console > /dev/console 2>&1
kmk:3:wait:/sbin/kmknod > /dev/console 2>&1
s2:23:wait:/sbin/rc2 < /dev/console > /dev/console 2>&1
s3:3:wait:/sbin/rc3 < /dev/console > /dev/console 2>&1
cons:1234:respawn:/usr/sbin/getty console console vt100
```

- Die ersten Zeilen im obigen Beispiel sind Kommentare.
- Danach folgt als erste Code Zeile die definition des default Runleves.
- Die Befehle /sbin/rc0, /sbin/rc2 und /sbin/rc3 starten jeweils die dazugehörigen scripts. es wird dabei der *distributed startup mechanism* verwendet. Es werden also nicht einzelne Dateien sondern gleich alle Dateien eines Directories gestartet (siehe 2.2.2).
- Das Kommando `update` führt ein regelmässiges `sync` auf den Platten durch. Das soll gewährleisten, dass das System bei einem Crash keine Daten verliert.
- Das Kommando `bcheckrc` führt den Filesystem Check mittels `fsck` durch und mountet die Filesysteme.

- Das Kommando `kmknod` prüft den laufenden Kernel ob alle benötigten Device Files (siehe 5.1) existieren. Dazu müssen die statisch deklarierten *Kernel Layered Products* mit `kreg` registriert werden.
- Das Kommando `kloadsrv` ist der Kernel Load Server. Er lädt die benötigten Module des Kernels.
- `it` schliesslich konfiguriert das System indem es Programme aus /sbin/it.d/bin aufruft, die Information wie passwd, timezone, etc. über das System sammeln.

2.2.2 startup files

Im Betriebsystem OSF/1 gibt es keine Scripts `rc` oder `rc.local` wie bei BSD Systemen. Hier geschieht die Konfiguration des Systems über das /etc/rc.config file. Es enthält Daten wie Hostname, IP-Nummer oder Network-Device.

Alle weiteren Startup Files sind in /sbin/rc.0, /sbin/rc.2 und /sbin/rc.3 zu finden. Die in diesen Verzeichnissen liegenden Scripts werden durch /sbin/rc0 /sbin/rc2 /sbin/rc3 einzeln aufgerufen. Nach der Konvention beginnen sie entweder mit einem 'K' (für Kill) oder einem 'S' (für Start) gefolgt von einer zweistelligen Ziffer und einem Namen.

```
K00enlogin    K05lpd    S60cron    S30nfs
```

Die Nummer ist wichtig da die Scripts nach diesen sortiert in aufsteigender Reihenfolge aufgerufen werden. Der Name dient lediglich der besseren Zuordnung der Scripts.

`rc2` führt dabei folgende Funktionen aus:

- Zeit Zone setzen
- Prüfen ob der Runlevel gewechselt wird
- Netzwerk Service und Daemons stoppen
- starten bzw. stoppen von System Daemons
- `cron` Daemon starten
- Konfigurieren der `dump`- und `paging` Tools

Manche dieser Funktionen werden nur beim Booten andere wieder nur beim Wechseln des Runlevels ausgeführt.

`rc3` wechselt schliesslich direkt in den Multiuser Status. Es führt zusätzlich zu den im Script selber definierten Kommandos auch alle Scripts des Verzeichnis /sbin/rc3.d aus. Die wichtigsten Aufgaben sind:

- 'run level' prüfen
- Netzwerk Service und Daemons starten
- System Service und Daemons starten
- und Filesysteme mounten

schliesslich gibt es noch `rc0` dieses hat folgende Aufgaben:

- Warnen der User, dass ein System Shutdown durchgeführt wird.
- Platten syncronisieren

- Stoppen aller System Services und Daemons
- Stoppen aller Netzwerk Services und Daemons
- Stoppen der Prozesse
- Unmounten der Filesysteme
- `init` aufrufen falls ein Shutdown zum Single User Mode durchgeführt werden soll

Das Stoppen aller Prozesse wird mit dem Kommando `killall` durchgeführt, dass jedem Prozess ausser jenen Prozessen, die dem Users gehören, welcher `killall` aufruft, ein `SIGTERM` gefolgt von einem `SIGKILL` sendet. Die Filesysteme bis auf das Root Filesystem werden mittels `umount -a` abgehängt.

Die Scripts in /sbin/rc0.d, /sbin/rc2.d und /sbin/rc3.d sind normalerweise Links auf Scripts im /sbin/init.d Verzeichnis. Diesen können die Parameter start, stop und meist auch restart übergeben werden. So startet man den sendmail service einfach mit /sbin/init.d/sendmail start. Ebenso kann man einen Restart mittels /sbin/init.d/sendmail restart durchführen.

```
#!/sbin/sh
PATH=/sbin:/usr/sbin:/usr/bin
export PATH

case "$1" in
'start')
        set `who -r`
        if [ "$9" = "S" ]; then
                if [ -f /usr/sbin/sendmail ]; then
                        (cd /usr/spool/mqueue; rm -f lf*)
                        /usr/sbin/sendmail -bi
                        /usr/sbin/sendmail -bd -q15m -om &
                        echo "SMTP Mail Service started"
                else
                        echo "Unable to start SMTP Mail Service, No /usr/sbin/sendmail"
                        exit 1
                fi
        else
                echo "Unable to start SMTP Mail Service, Run level must be S"
                exit 1
        fi
        ;;
'stop')
        pid=`/bin/ps -e | grep sendmail | grep connection | sed -e 's/^  *//' -e's/ .*//' | head -1`
        if [ "X$pid" != "X" ]; then
                /bin/kill $pid
        else
                echo "No pid for SMTP Mail Service found"
                exit 1
        fi
        ;;
'restart')
        /sbin/init.d/sendmail stop
        if [ $? -ne 0 ]; then
                echo "  -- Could not find running sendmail - attempting to restart..."
        fi
        sleep 2
```

```
        pid=`/bin/ps -e | grep sendmail | grep connection | sed -e 's/^  *//' -e's/ .*//' | head -1`
        if [ "X$pid" != "X" ]; then
                sleep 3
                pid=`/bin/ps -e | grep sendmail | grep connection | sed -e 's/^ *//' -e 's/ .*//' | head -1`
                if [ "X$pid" != "X" ]; then
                        echo "Could not kill SMTP Mail Service"
                        echo "  -- Kill sendmail by hand then do a start"
                        exit 1
                fi
        fi
        /sbin/init.d/sendmail start | sed -e 's/start/restart/'
        ;;
*)
        echo "usage: $0 {start|stop|restart}"
        ;;
esac
```

Das Beispiel zeigt das Script `sendmail`; es besteht grob aus der Struktur:

```
case $1 in
`start`) commands....
         ;;
`stop`) commands....
        ;;
`restart`) commands....
           ;;
*) usage meldung falls kein parameter angegeben
   ;;
esac
```

Diese Struktur sollte von jedem Script in diesem Verzeichnis befolgt werden da die System Scripts `rc0` `rc2` und `rc3` diese Konventionen für den Start bzw. für das Stopen der Prozesse verwenden.

2.2.3 runlevel ändern

Beim Booten geschieht das Ändern des Runlevels automatisch. Es ist aber auch sinnvoll den Runlevel von Hand zu ändern. Vor allem dann wenn man am System Änderungen durchführt. Dazu wechselt man mit `init [0sS23]` in den gewünschten Runlevel. So kann auch ein Shutdown des Systems mit `init 0` vollzogen werden oder mit `init s` ein Wechsel in den Single User Mode (siehe 2.4).

```
                                          shutdown -h time   shutdown time
 4-9  Frei definierbare 'run level'              |                |
                                                 |                |
 3    Multiuser Mode mit Netzwerk Services       |                |
                                                 |                |
 2    Multiuser Mode mit Lokalen Services        |                |
                                                 |                v
 s,S  Single User Mode (nur root Filesystem)     |              halt
                                                 |                |
 0    boot Console                               v                v
```

2.3 shutdown

Der Shutdown des Systems kann auf zwei Stufen erfolgen (siehe auch 2.2.3). In den Single User Mode und weiter auf die Boot-Konsole (siehe 2.5). Dazu steht das Kommando `shutdown` zur Verfügung.

```
syntax: /usr/sbin/shutdown [-bfhknr] time [warning-message ...]
```

Das Argument `time` muss dabei immer angegeben sein. Man kann das Schlüsselwort `now` verwenden um den Shutdown sofort einzuleiten oder mit `+number` bzw. mit `hhmm` die Zeit in Minuten ab dem Start bzw. in Stunden (hh) und Minuten (mm) angeben. Die Flags bedeuten:

-b

Die 'warning message' wird an den `rwalld` daemon geschickt, so dass alle remote clients, die per NFS Filesysteme gemountet haben, auch die Nachricht erhalten.

-f

fastboot, vesucht das System so schnell wie möglich herunter zu fahren. Die Meldung an die User wird übergangen.

-h

halt, fährt das System ganz herunter - bis auf die Boot-Konsole.

-k

warnt die User vor einem Shutdown führt diesen allerdings nicht durch.

-n

führt den Shutdown ohne Syncronisieren der Disks durch.

-r

veranlasst dass das System nach dem Shutdown automatisch bootet. Das System bleibt nicht in der Boot-Konsole stehen.

Während des Shutdowns wird in /etc die Datei nologin angelegt. Sie enthält die 'warning message' und wird beim Versuch sich in das System remote Einzuloggen angezeigt. Für die meisten der oben genannten Flags gibt es auch Aliases: `fastboot` für `shutdown -f now`, `reboot` für `shutdown -r now`, und `halt` für `shutdown -h now`

2.4 Single User Mode

Wichtig im Betrieb eines Rechners ist auch der Single User Mode. In diesem Modus stehen bereits die wichtigsten Programme zur Systemadministration zur Verfügung. Er kann dazu verwendet werden ein 'gecrashtes' System zu reparieren oder eine System- oder Kernelkonfiguration vorzunehmen (siehe auch 5.2). In den Single User Mode kommt man mit `shutdown time` (siehe 2.3) oder mit `init s` (siehe 2.2.3). Nicht immer ist es möglich vom Multiuser Level in den Single User Mode zu gelangen. Nach einem Crash kann es möglich sein, dass sich das System nicht in den Multiuser Level starten lässt. Oft bleibt es dann schon im Single User Mode stehen, was unser Problem, in das System zu kommen und es zu warten, erleichtert. Hie und da bleibt es aber erst nach dem Single User Mode stecken. Dann hilft nur ein modifiziertes Boot Kommando von der Boot-Konsole `boot -FL S` aus. Mit diesem Kommando kommen wir von der Boot-Konsole aus in den Single User Mode.

2.4.1 fsck

Sehr oft kommt es vor, dass nach einem System Crash die Filesysteme von Hand zu prüfen sind. Dabei bleibt das System meist automatisch im Single User Mode stehen und fordert uns dazu auf. Es wird dabei ausgegeben welches Filesystem von einem nicht automatisch reparierbaren Fehler betroffen ist. Dieser kann dann durch manuelles aufrufen von `fsck` behoben werden.

```
syntax: /usr/sbin/fsck [fs_options] [filesystem...]
```

Optionen für `fsck`:

-b block

verwendet den spezifizierten Block als alternativen Superblock (siehe auch 4.3) für das Filesystem. Block 32 ist normalerweise ein alternativer Superblock.

-c

konvertiert das alte Format mit statischen Tabellen in das neu mit dynamischen Tabellen und zurück. Wenn nicht das Flag `-p` zusätzlich angegeben ist wird der Operator zuerst gefragt ob die Konvertierung durchgeführt werden soll.

-l number

gibt an wieviele parallele checks durchgeführt werden können.

-m mode

verwendet den angegebenen `mode` als Oktal Permission Bits bei der Erstellung des `lost+found` Verzeichnis.

-n

gibt allen Fragen bis auf die Frage '`CONTINUE?`' 'no' zurück.

-p

schaltet den interaktiven Mode aus und korrigiert folgende Filesystem Fehler: unreferenzierte Inodes *unreferenced inodes*, zu grosse Link Counts in Inodes *link counts in inodes that are too large*, fehlende Blöcke in der Free Map *missing blocks in the free map*, Blöcke in der Free Map, die ebenso in Dateien auftauchen *blocks in the free map that are also in files*, und Falsche Counts in den Super Blocks *wrong counts in the super-block*.

-y

gibt an jede Frage des Programms 'yes' zurück.

2.5 bootconsole

In der Bootkonsole können einige Parameter, die das Verhalten der Maschine nach dem Einschalten betreffen, verändert werden. So kann das Boot Device, das automatische Booten beim Einschalten, die Keyboard Language und vieles mehr verändert werden. Nachfolgend werden die wichtigsten Konsole Kommandos vorgestellt.

2.5.1 Das boot Kommando

Das `boot` Kommando initialisiert das System und startet das boot Programm, welches den Kernel ladet und startet (siehe 2.1.1).

```
boot <-flags> <-filename> boot_device
```

-fl value

Das wichtigste Flag 'S' haben wir bereits beim booten von der Konsole in den Single User Mode gehabt, `boot -fl S` bootet den Rechner von der Boot-Konsole in den Single User Mode und bleibt auf diesen 'run level'.

-fi filename

Mit diesem Parameter kann dem `boot` Kommando der Name der Datei übergeben werden, die in das System geladen werden soll. `boot -fi "alternativ_boot"`

boot_device

Das Device von dem das System booten soll. (z.B.: `rz1a`) Dabei unterscheiden sich die Systeme. Die genaue Syntax finden Sie im Owners Guide. Für die TURBOchannel Devices gilt zum Beispiel folgende syntax:

```
boot "#/device" <-flags> <-filename>
```

Wobei `#` die Slot Nummer und das `device` das Boot Device (z.B.: boot "2/DKA200") ist.

2.5.2 Das show Kommando

Das `show` Kommando testet das System und zeigt den gefundenen Status an der Konsole an. Es gibt vier mögliche Parameter.

```
show parameter
```

show config

zeigt die System Komponenten, die Revision Nummer und die installierten TURBOchannel Optionen an.

```
show config

DEC 3000 - M800
Digital Equipment Corporation
     VPP PAL X5.41-82000101 - Built on 10-MAY-1993 00:00:00.00

TCINFO     DEVNAM       DEVSTAT
------     --------     ---------
                CPU     OK KN17-AA-V3.0-S086-I062-DECchip 21064 P3.0
               ASIC     OK
                MEM     OK
8
7
                NVR     OK
                SCC     ? 60
                 NI     OK
               ISDN     OK
6
               SCSI     OK
1-PMAGB-BA      TC1     OK
```

Die Ausgabe des Kommandos `show config` hat drei Spalten.

TCINFO

enthält die TURBOchannel Device Information. Die Information neben der TCx Komponente zeigt an welche TURBOchannel Option installiert ist und welchen Slot sie belegt.

DEVNAM

Name der System Komponente oder Modules. Die wichtigsten Komponenten folgen

```
CPU     Processor
ASIC    Application specific integrated circuit
MEM     Memory
NVR     Non volatile RAM and time-of-year (TOY) clock
NI      Network Interface (Ethernet)
SCC     Serial communications controller
SCSI    SCSI Interface
TCx     TURBOchannel (x ist die Nummer des TURBOchannel Devices)
ISDN    ISDN Anschluss und Audio Port
```

DEVSTAT

Status der Komponente. Ein Fehler wird durch ein Fragezeichen Symbolisiert. Im obigen Beispiel sehen wir es neben dem Eintrag für die Serielle Schnittstelle (SCC).

Bei Maschinen mit neuem BIOS und PCI Hardware ergibt sich ein etwas geändertes Bild. Das Display enthält nach Eingabe von `show config` folgende Informationen:

- Versionsnummer der Firmware, des PALcode, des SROM Chips und der CPU
- Die Grösse und Konfiguration für jede Memory-Bank
- PCI Bus Informationen
- EISA Bus Informationen

```
>>> show config
Firmware
SRM Console: X4.4-5365
ARC Console: 4.43p
PALcode:     VMS PALcode X5.48-115, OSF PALcode X1.35-84
Serial ROM:  X2.1

Processor
DECchip (tm) 21064A-6 266

MEMORY
        32 Meg of System Memory
        Bank 0 = 32 Mbytes (8 MB Per Simm) Starting at 0x00000000

PCI Bus
    Bus 00 Slot 07: Intel 8275EB PCI to EISA Bridge

    Bus 00 Slot 08: Digital PCI to PCI Bridge Chip

      Bus 02 Slot 00: ISP1020 Scsi Controller
                      pka0.7.0.2000.0    Scsi Bus ID 7
                      dka0.0.0.2000.0    RZ29B
                      dka400.4.0.2000.0  RRD45

      Bus 02 Slot 04: DECchip 21040 Network Controller
                      ewa0.0.0.2004.0  08-00-2B-E5-6A-41

EISA Bus Modules (installed)
>>>
```

Dabei gelten folgende Bezeichnungen für die Devices:

- dr–RAID set device

- dv–Floppy drive
- er–Ethernet port(LANCE chip, DEC 4220)
- ew–Ethernet port(TULIP chip, DECchip 21040)
- pk–SCSI port
- dk–SCSI disk
- mk–SCSI tape
- pu–DSSI port
- du–DSSI disk
- mu–DSSI tape

Der nachfolgende Buchstabe gibt den Adapter an (a,b,c...) dann folgt die Device Nummer (bei SCSI *100 zB.: dka400), die Bus Node ID, die Channel Nummer, die Logical Slot Nummer und die Hose Nummer (0=PCI, 1=EISA) Diese Art der Bezeichnung findet man auch bei den nachfolgenden Kommandos.

show device

zeigt die vorhandenen Devices an wie: Ethernet Interface, SCIS Tapes, interne als auch externe SCSI disk Drives und sonstige SCSI Geräte.

```
show device

BOOTDEV    ADDR      DEVTYPE      NUMBYTES   RM/FX   WP   DEVNAM   REV
-------    ----      -------      --------   -----   --   ------   ---
ESA0       08-00-2B-12-00-9C , THICK
DKA100     A/1/0     DISK           1.05GB   FX           RZ26     T368
DKA400     A/4/0     RODISK          ....    RM      WP   RRD42    4.3d
..HostID.. A/7       INITR
..HostID.. B/7       INITR
```

BOOTDEV

Name des Boot Devices. In diesem Fall das CDROM am BUS A mit der SCSI ID 4

ADDR

Interne Adresse des Devices, wie im Beispiel, die der Ethernet Karte

DEVTYPE

Typ des Devices: Disk, Interface etc.

NUMBYTES

Kapazität der Platten in MBytes

RM-FX

Disk Type, RM für Removeable und FX für Fixed

WP

Write Protected

DEVNAM

Device Name, normalerweise der Digital Produktname (z.B.: RZ26)

REV

Revision Nummer

show error

zeigt jeden gefundenen Fehler des Systems an

```
show error

? 003    SCC  0x0050
  ? T-ERR-SCC-LK401 - 0 char rcvd
     T-STS-SCC-LK401 - char = 700
```

?

Fehler Indikator

003

FRU (Field Replaceable Unit) Nummer. Jede Komponente oder Modul im System hat eine eigene Nummer.

SCC

Der Name des Diagnostic Tests, der den Error gefunden hat.

0x0050

Die Fehlernummer im Hexadezimal Format

Alle weiteren Fehlermeldungen sind für uns nicht interpretierbar sondern sollten an den Digital Techniker weitergemeldet werden.

Bei neueren Maschinen verwendet man `cat el` oder `more el` um die Fehlermeldungen anzuzeigen.

show memory

zeigt alle Memory Module an

```
show memory

DEC 3000 - M800 Memory: 128 Mbytes
------------------------------------------
BANK #      MEMORY_SIZE        START_ADDRESS
------      -----------        -------------
  0         032 Mbytes         0x00000000000
  1         032 Mbytes         0x02000000000
  2         032 Mbytes         0x04000000000
  3         032 Mbytes         0x06000000000
  4         000 Mbytes         0x00000000000
  5         000 Mbytes         0x00000000000
  6         000 Mbytes         0x00000000000
  7         000 Mbytes         0x00000000000
```

Obiges Beispiel zeigt dass die Memory Bänke 0,1,2 und 3 mit jeweils 32 MB Module bestückt sind. Die Bänke 4-7 sind frei.

2.5.3 Das test Kommando

Mit dem `test` Kommando können die oben bereits erwähnten Komponenten (siehe 2.5.2) getestet werden.

```
test component

component:

ASIC Application specific integrated circuit
```

```
MEM  Memory
NVR  Non volatile RAM and time of year (TOY) clock
SCC  Serial communications controller
NI   Network interface (Ethernet)
SCSI SCSI devices
ISDN ISDN und Audio Port
TCx  TURBOchannel
```

2.5.4 Das printenv Kommando

Zeigt alle oder einzelne System Variablen an.

```
printenv auto_action

AUTO_ACTION = HALT
```

2.5.5 Das set Kommando

Mit `set variable parameter` können folgende System Variablen geändert werden.

```
Variable         Beschreibung                                  Default
--------------------------------------------------------------------
auto_action      Spezifiziert die Aktion, die die Konsole      HALT
                 ausfuehren soll nachdem das System
                 eingeschalten wurde.
                 restart, boot und halt sind die moeglichen
                 Werte.
bootdef_dev      default Boot Device                           {null}
                 mit show device kann man die n\"otigen Daten
                 herausfinden
boot_osflag      Boot Flags                                    0,0
boot_reset       gibt an ob das System vor dem Booten          ON
                 einen Reset durchfuehren soll
ethernet         Setzt den Typ des Ethernet THICK oder TENBT   THICK
language         Konsole Keyboard Mappings
server           Server Status                                 OFF
                 gibt an ob der Rechner als Server oder
                 als Workstation arbeitet. Als Server kann
                 er auch ohne Konsole gefahren werden, dann
                 muss diese Variable auf ON geschalten werden.
```

Die Tabelle zeigt nur die wichtigsten System Variablen, weitere Variablen finden Sie im Owners Guide.

3 SCSI-Standard: Small Computer System Interface

Dieser Standard erlaubt es Platten oder Geräte anderer Hersteller problemlos ins System zu integrieren. Der Standard besitzt folgende Charakteristik:

- Definition eines gemeinsamen Interface, das es erlaubt Produkte verschiedener Hersteller einzusetzten.
- Festlegung der Kommunikation mit dem Rechner unabhängig vom herstellerabhängigen Gerätelayout.

- Es können verschiedenste Geräte wie Platten, Scanner, Drucker und Bandlaufwerke angeschlossen werden.
- Jedes Gerät hat eine eindeutige ID Nummer

Achtung! auch der Hostadapter hat eine ID Nummer, in der Regel die Nummer 7. Sie darf natürlich nicht nocheinmal für ein Gerät verwendet werden.

Zum Absuchen einer Platte nach fehlerhaften Sektoren kann man das Tool `scu` verwenden. **Achtung!** es zerstört den Inhalt der Platte.

4 Harddisk und Filesystem

In diesem Kapitel soll die Konzepte zur Installation von Festplatten erörtert werden. Um dies auch, speziell in Hinsicht auf das Partitionieren von Festplatten, zu verstehen soll zuerst einmal auf den Physikalischen Aufbau von Festplatten eingegangen werden.

4.1 Physikalischer Aufbau

Eine typische Festplatte hat einen zylindrischen Aufbau. Über Magnetköpfe werden die Daten auf die rotierenden Platten geschrieben oder von ihnen gelesen. Die Platte ist dabei in verschiedene Einheiten unterteilt, die mit nachfolgenden Begriffen bezeichnet werden.

tracks

Dies sind die Positionen, die die Schreib/Lese Köpfe auf der Platte einnehmen können. Sie sind unterteilt in

sectors

Die kleinste logische Einheit der Platte, und in der Regel 512 Bytes gross.

cylinder

Zylinder fassen Gruppen von Tracks zusammen, die den gleichen Abstand zum Plattenzentrum haben.

4.2 Partitionieren

Das Partitionieren der Festplatte erfolgt mit dem Befehl `disklabel -e device` . Soll das Default Partition Layout für einen Platten Typ übernommen werden so kann man dies mit dem Befehl `disklabel -w -r` dev/rrz2a RZ28/ tun. Dies funktioniert nur dann, wenn der Disktyp in der Datei /etc/disktab vorhanden ist. Ansonst muss man den neuen Typ erst in der Datei `disktab` eintragen.

```
# ein Beispiel Eintrag aus /etc/disktab
rz28|RZ28|DEC RZ28 Winchester:\
        :ty=winchester:dt=SCSI:ns#99:nt#16:nc#2595:\
        :oa#0:pa#131072:ba#8192:fa#1024:\
        :ob#131072:pb#401408:bb#8192:fb#1024:\
        :oc#0:pc#4110480:bc#8192:fc#1024:\
        :od#532480:pd#1191936:bd#8192:fd#1024:\
        :oe#1724416:pe#1191936:be#8192:fe#1024:\
        :of#2916352:pf#1194128:bf#8192:ff#1024:\
        :og#532480:pg#1787904:bg#8192:fg#1024:\
        :oh#2320384:ph#1790096:bh#8192:fh#1024:
```

folgende Variablen stehen dabei zur Verfügung um eine Platte und deren default Disklabel zu definieren - nicht alle müssen vorhanden sein. d(0-4) ist als d1,d2,d3,d4 zu verstehen ebenso b[a-h] als ba,bb,bc...,bh usw.

ty string

Type der Platte (z.B.: removable, winchester)

dt string

Controller Typ (SCSI,ESDI..)

ns nummer

Sectoren pro Track

nt nummer

Tracks pro Zylinder

nc nummer

Anzahl der Zylinder - gesamt

sc nummer

Sectoren pro Zylinder (nc*nt)

su nummer

Anzahl der Sectoren - gesamt (sc*nc)

se nummer

Bytes pro Sector

sf bool

Controller unterstützt bad144-style bad-sector-forwarding

rm nummer

Rotationsgeschwindigkeit in rpm

sk nummer

Sector Skew pro Track, default 0

cs nummer

Sector Skew pro Zylinder, default 0

hs nummer

Headswitch time, default 0

ts nummer

One-Cylinder seek time, default 0

il nummer

Sector Interleav, default 1

d(0-4) nummer

Plattenspezifische Parameter

bs nummer

Grösse des Bootblocks

sb nummer

Grösse des Superblocks

b(a-h) nummer

Blockgrösse der Partition [a-h]

f(a-h) nummer

Fragmentgrösse der Partition [a-h]

o(a-h) nummer

Offset der Partition [a-h] in Sectoren

p(a-h) nummer

Grösse der Partition [a-h] in Sectoren

t(a-h)=string

Typ der Partition (z.b.: 4.2BSD)

```
# disklabel -r /dev/rrz2a
# /dev/rrz2a:
type: SCSI
disk: rz28
label:
flags:
bytes/sector: 512
sectors/track: 99
tracks/cylinder: 16
sectors/cylinder: 1584
cylinders: 2595
sectors/unit: 4110480
rpm: 3600
interleave: 1
trackskew: 0
cylinderskew: 0
headswitch: 0           # milliseconds
track-to-track seek: 0  # milliseconds
drivedata: 0

8 partitions:
#        size   offset    fstype   [fsize bsize   cpg]
  a:   131072        0    4.2BSD     1024  8192    16   # (Cyl.    0 - 82*)
  b:   873813   131072    unused     1024  8192         # (Cyl.   82*- 634*)
  c:  4110480        0    unused     1024  8192         # (Cyl.    0 - 2594)
  d:   873813   131072    unused     1024  8192         # (Cyl.   82*- 634*)
  e:   873813   131072    unused     1024  8192         # (Cyl.   82*- 634*)
  f:   873813   131072    unused     1024  8192         # (Cyl.   82*- 634*)
  g:  3105595  1004885    4.2BSD     1024  8192    16   # (Cyl.  634*- 2594*)
  h:   873813   131072    unused     1024  8192         # (Cyl.   82*- 634*)
```

Aus obigen Output einer Digital Festplatte (Typ: RZ28) ergibt sich aus

```
(cylinders) * (tracks/cylinder) * (sectors/track) * (bytes/sector)/(1024*1024) =
                    1584 * 16 * 99 * 512 / (1024 * 1024) = 2007MB
```

In der Tabelle partitions bedeutet

size

Die Grösse der Partition in Sectoren.

offset

Der Offset zum 0 sector

fstype

Der Filesystemtyp mit dem die Partition formatiert ist.

möchte man jetzt eine Partition mit 64MB einrichten kann man sich die Grösse der Partition in Sectoren folgendermassen ausrechnen:

```
(Groesse der Partition in MB) * 1024 * 1024 / (bytes/sector) =
64 * 1024 * 1024 / 512 = 131072 sectoren
```

Die Anzahl der Zylinder dieser Partition bekommt man aus

```
(Anzahl der Sectoren) / ( sectors/track) / (tracks/cylinder)  =
131072 / 99  / 16 = 82
```

Diese Zahl sehen wir auch im Kommentar zur ersten Zeile in der Partition Tabelle. Dabei wird von Cylinder 0 an gezählt. Die Kommentarzeile kann man lesen als: die Partition reicht von Zylinder 0 bis zum Cylinder 82. Achtung der Zylinder 82 ist nicht inkludiert, was der '*' auf der rechten Seite anzeigt.

Die Grösse der Platte in Sectoren und damit die grösstmögliche Partition der Platte errechnet man sich aus

```
(cylinders) * (tracks/cylinder) * (sectors/track)
2595 * 99 * 16 = 4110480
```

Hat man eine Offset so errechnet sich der oberste Zylinder aus

```
(size + offset ) / (tracks/cylinder) / (sectors/track)
(131072+873813) / 99 / 16 = 634
```

der untere Zylinder schliesst an der vorigen Partition an.

Eine typische OSF/1 Partitionstabelle schaut wie folgt aus:

```
0...........................max sector
                 [   d    ][ e ][ f ]
[  a   ][   b   ][      g     ][  h  ]
[                c                   ]
```

Das Ändern in einer Zeile der Partitionstabelle hat keine Auswirkung auf die nebenstehenden Einträge. So können sich mehrere Einträge des Disklabels überlappen. Das hat so lange keinen Einfluss so lange man nicht versucht auf sich überlagernde Partitions Filesysteme zu installieren.

4.3 Filesystem

Während die Daten auf dem physikalischen Datenträger bitweise abgelegt sind, wird durch die Einführung des Filesystemkonzeptes eine für den Benutzer transparente Zuordnung der Daten vorgenommen. Der Systemkern implementiert das Filesystem und übernimmt so die Kontrolle und den Zugriff auf die Daten. Hierzu besitzt der Systemkern zu jedem Filesystem, das im Systembaum eingehängt ist, einen *Superblock*, der den Aufbau des jeweiligen Filesystems kennzeichnet. Zudem verwaltet der Kernel eine *Inodetabelle*, die unter anderem für jedes File Informationen über Zugriffsrecht, Inhaber und Lokalisierung der Daten auf der Platte enthält. Das Filesystem ist also eine Abstraktionsebene des Kernels und ermöglicht dem Benutzer den einfachen Zugriff auf die Daten.

4.3.1 Elemente des Filesystems

Jedes Filesystem enthält folgende Elemente:

- Einen Superblock, der über den grundsätzlichen Aufbau des Filesystems Auskunft gibt und die Grösse der verwendeten Namen, die Liste der freien Inodes, List auf freie Datenblöcke, Modifikationsdatum etc. enthält. Hiervon sind mehrere Kopien auf der Festplatte angelegt. Der erste alternative Superblock ist auf Block 32 zu finden. Ist das Filesystem gemountet so wird der Superblock in den Kernel geladen und mittels `update` oder `sync` werden die Kopien auf der Festplatte mit der Version des Kernels abgeglichen.
- Eine Inode Tabelle. In ihr existieren zu jeder Datei die Informationen über Zugriffsrechte, Lokalisierung der Daten auf der Platte etc.
- Den Datenbereich. In ihm werden schliesslich die Daten abgelegt.

Das Betriebsystem OSF/1 verwendet per default das BSD Filesystem.

```
 ------------ -------- -------------- ----------- -------- ------- ---
|            |        |              |           |        |       |
| Boot Block | Super  | Alternativer | Cylinder  | Inode  | Data  | next
|            | Block  | Superblock   | Block     | Table  |       | Cylinder
|            |        |              |           |        |       | Group
|            |        |              |           |        |       |
 ------------ -------- -------------- ----------- -------- ------- ----
                           \                                 /
                            --------------\  /---------------
                                           \/
                                     Cylinder Group
Partition:
 [Boot Block][Superblock][Cylinder Group1]....[Cylinder GroupN]
```

Der Vorteil liegt in

- den langen Dateinamen (256 Zeichen)
- den schnellen Zugriff auf die Daten durch erhöhte Blockgrösse
- Platzersparnis durch Fragmentierung
- Einführung von Zylindergruppen ermöglicht schnelleren Datenzugriff da die Daten und Inode Table zu kleineren Gruppen zusammengefasst werden.
- die Alternativen Superblocks erhöhen die Sicherheit.

Die Ausgabe der Blöcke in denen die Alternativen Superblöcke gespeichert sind erreicht man über den Befehl `newfs -N special_device`, dabei wird kein Filesystem auf der Platte erzeugt sondern lediglich deren Filesystemparameter ausgegeben.

4.3.2 Filesysteme installieren

Das Filesystem installiert man mit dem Befehl `newfs`

```
syntax: newfs [-N] [newfs-options] special-device [disk-type]
```

Die Bedeutung der Flags können Sie mit `man newfs` nachlesen. Sie werden eher seltener gebraucht. Mit

```
# newfs /dev/rz2a

/dev/rrz2a:     131072 sectors in 83 cylinders of 16 tracks, 99 sectors
        67.1MB in 6 cyl groups (16 c/g, 12.98MB/g, 3008 i/g)
super-block backups (for fsck -b #) at:
 32, 25488, 50944, 76400, 101856, 127312,
```

können wir jetzt die Partition a mit einem Filesystem versehen um sie danach mittels

```
mount /dev/rz2a /mountpoint
```

zu mounten.

4.3.3 Filesysteme mounten

Das Einhängen des neuen Filesystems in den vorhandenen Filesystembaum geschieht über den Befehl

```
syntax: mount [options] [device-file] [mountpoint]
```

Optionen:

-a

mountet alle Filesysteme die im File /etc/fstab vermerkt sind

-t filesystem

spezifiziert ein spezielles Filesystem wie `ufs` oder `nfs`

-r

mountet das Filesystem 'read only'

-o optionen

Filesystem spezifische Optionen.

-l

zeigt alle Informationen zu den vorhandenen Filesystemen

Analog dazu erfolgt das Entfernen des Filesystems aus dem Systembaum mit dem Befehl:

```
umount [options] file-system
```

Beispiele:

1. Mounten der Partition(g) einer lokalen Platte mount /dev/rz2g /usr
2. Mounten aller Filesysteme ausser jenen mit dem Typ nfs `mount -at nonfs`
3. Mounten eines NFS Filesystems vom Server ftp mount -t nfs ftp:/home/ftp/pub /mnt
4. mounten des CDROM Laufwerks mount -r /dev/rz4c /cdrom
5. mounten einer CD mit dem ISO9660 Filesystem ohne Versionsnummern mount -t cdfs -r -o noversion /dev/rz4c /cdrom

Sollen Filesysteme bereits beim Booten gemountet werden so muss man sie in die Datei /etc/fstab eintragen.

```
# cat /etc/fstab
# device      mountpoint   Typ     Zugriff
#                                  (readwrite,swap,readonly) backup    fsck
#
/dev/rz2a       /          ufs     rw                        1         1
/proc          /proc       procfs  rw                        0         0
/dev/rz2g      /usr        ufs     rw                        1         2
/dev/rz2b      swap1       ufs     sw                        0         2
/dev/rz3c      /al03_1     ufs     rw                        1         2
/dev/rz6c      /al03_2     ufs     rw                        1         2
```

Dabei gibt das Feld fsck an in welcher Reihenfolge die Filesysteme überprüft werden sollen (0=kein fsck). Das Feld backup gibt dem Programm `dump` an welches Filesystem zu sichern ist. Im Feld Zugriff können die Zugriffsrechte und Zugriffsarten festgelegt werden. Dabei steht

rw

read write

rq

read write und quotas

sw

Swap Device

xx

Diese Zeile soll von mount ignoriert werden

4.3.4 Filetypen und Filepermissions

Zum Abschluss der Beschreibung des Filesystems sollen noch einmal die Filetypen und Filepermissions in Erinnerung gerufen werden. Sehr wichtig fuer den Sytemadministrator sind die Filepermission über die der Zugriff auf Files in einem Filesystem geregelt werden kann.

Die Filepermissions bestehen aus einer vierstelligen Octalzahl. Wobei die erste Octalzahl spezielle Filepermission setzt. Die 2,3 und 4 geben die Filepermission für den Eigentümer, die Gruppe und den Rest der Welt an. Die Werte dieser Octalzahlen haben folgende Bedeutung:

- 0 keine Rechte (—)
- 1 exekutierbar (–x)

- 2 Schreibrechte (-w-)
- 3 Schreib und exekutierbar (-wx)
- 4 Leserechte (r–)
- 5 Leserechte und ausführbar (r-x)
- 6 Schreib- und Leserechte (rw-)
- 7 Schreib-, Leserechte und ausführbar (rwx)

Wie schon erwähnt hat die erste Octalzahl eine Sonderstellung. Mit ihr kann das SUID Bit gesetzt werden. Man unterscheidet dabei folgende Rechte:

SUID (octal=4000)

Ist dieses Mode gesetzt und ist das Programm ausführbar so läuft es unter der User ID des Inhabers und nicht unter der des aufrufenden Benutzers. Dies gestattet beispielsweise das auch nichtpriviligierte Benutzer ihr Passwort ändern dürfen da man beim Zugriff auf die Datei /etc/passwd Root Rechte braucht. SUID eröffnet damit aber auch beim unvorsichtigen Einsatz alle Türen des Systems. **Nie darf ein Shell Script von `root` mit dem SUID versehen werden!!!** Der einfache Aufruf der Escape-Sequenz gibt dem aufrufenden Benutzer vollständige Superuser Privilegien.

SGID (octal=2000)

Wie das SUID allerdings auf die Gruppe angewandt. Programme laufen dann in der Gruppe die dem Programm zugeordnet ist.

sticky bit (octal=1000)

Der einstige Grund das 'sticky bit' einzuführen war, dass das programm, welches mit diesem versehen war, sich permanent im Hauptspeicher des Rechners festsetzte. Diese Option ist mittlerweile überholt. Heute dient das 'sticky bit' dazu ein Verzeichnis für alle Benutzer schreibbar zu machen ohne dass Benutzer die Dateien eines anderen Benutzers beschreiben oder manipulieren können.

Obige Permissions können mit dem Befehl

```
chmod [-fR] absolut file
chmod [-fR] [ugo]+|-[rwxstX] file
```

gesetzt werden. Führt man eine Softwareinstallation durch und will man sicher sein, dass die richtigen Permission gesetzt sind so kann man mit `umask permmask` eine default Permission Mask vorgeben. Dabei bedeutet hier `permmask` nicht, dass dann die Files mit dieser Permission ausgestattet werden sondern dass sie genau diese nicht bekommen. Eine `umask` von 077 bedeutet, dass der Benutzereintrag alle Rechte die Group und die World keine Rechte bekommen. Eine `umask` von 022 bedeutet dass der User alle, die Group und World keine Schreibrechte bekommt.

All diese Modes können auf die verschiedenen Filetypen des UNIX Systems angewandt werden. Diese sind:

1. Regular Files: Benutzer- und Systemdaten, Programme und sonstige Plain Files.
2. Verzeichnisse: sie können im Listing am führenden 'd' erkannt werden `drwxr-x---`
3. Character und Block Device Files: Diese Files bilden die Benutzerschnittstelle zum Zugriff auf die Devices (siehe 5)

   ```
   crw------- Character Device
   brw-rw-rw- Block Device
   ```

4. Symbolic Links
5. Named Pipes
6. UNIX Sockets
7. hard linked Files, diese können nur duch ls -il erkannt werden. Nur die Ausgabe zeigt uns, dass beide die selbe Inode haben

```
ls -il
98675 -rw-rw-rw- 2 root root 1000 Dez 8 14:00 data
98675 -rw----rw- 2 root root 1000 Dez 8 14:00 data
```

4.4 Swap-Bereich

4.4.1 Grundkonzept

Das UNIX Betriebssystem unterstützt ein virtual memory Konzept. Dieses erlaubt es, die Grösse des physikalischen Hauptspeichers zu erweitern, indem Teile des Massenspeichers als Swap Space konfiguriert werden. Auf diesen werden Programmteile, die nicht mehr in den Hauptspeicher passen, ausgelagert. Durch Aufteilen der Programme in Pages ist es auch möglich Programme auszuführen, deren Grösse die des Hauptspeichers übersteigt. Dabei werden die Programme in Pages zerteilt, die bei Bedarf in den Speicher ausgelagert werden können.

Die Grösse des Swap Bereichs richtet sich nach der Grösse der Anwendungen, die auf der Maschine laufen sollen. Als Faustregel kann man das 2 fache des Hauptspeichers als Swap Bereich annehmen. Der Swap Bereich kann später durch Dazuhängen von Platten erweitert werden.

4.4.2 Konfiguration

Wird das Konzept der Plattenpartitionierung verwendet, so sind folgende Schritte zum Einrichten des Swap Bereichs durchzuführen:

1. Auswahl einer freien Partition und Überprüfung der dazugehörigen Device Files.
2. Den Swap Bereich mit `swapon device` aktivieren
3. Eintragen des Swap Bereichs im /etc/fstab
4. Prüfen ob der Swap Bereich ins System übernommen wurde mit `swapon -s`

Existiert der Link /sbin/swapdefault -> swap device so wird für jeden gestarteten Prozess automatisch hinreichend Swap Space reserviert. Dies kann zur Folge haben, dass keine neuen Prozesse mehr gestartet werden können, obwohl der Swap Bereich nicht völlig ausgelastet ist. Anderseits verhindert der Link, dass Prozesse plötzlich abbrechen da ihnen der virtuelle Speicher ausgeht.

5 Devices

Die Device Files bilden die Benutzerschnittstelle, die die Lese- und Schreibbefehle an den Kernel weiterreichen. Das System unterscheidet dabei *Block* und *Character Devices* und sucht aus einer internen Tabelle die zum jeweiligen Gerät passende Aktion aus.

Die Befehle werden schliesslich an die im System integrierten Device Treiber weitergeleitet und dort derart umgesetzt, so dass die einzelnen Geräte die Instruktionen auch ausführen können. Beim Anschluss einer neuen Platte müssen wir also diese sowohl im Kernel als Device Treiber als auch in der Benutzerschicht als Device File anmelden.

5.1 special files

Wie oben bereits angesprochen erfolgt der Zugriff des Benutzers auf die Geräte (wie Platten oder Bandlaufwerke) über das Filesystem. Dieses enthält im Verzeichnis /dev Special Files, die die Schnittstelle zum Benutzer bilden. Die Zuordnung der Special Files zu den Device Treiber erfolgt über eine Jumptable, wo jedes Special Files durch die *major* und *minor* Nummer klassifiziert wird. Dabei spezifiziert die *major* Nummer das Gerät wie zum Beispiel Tape, Platte oder Scanner. Die *minor* Device Nummer einen speziellen Typs des Geräts zum Beispiel bei einer Bandstation den Typ rewind und norewind.

5.1.1 Erzeugen von Special Files

Die Special Files befinden sich in der Regel im Verzeichnis /dev, wobei auch teilweise schon eine Unterteilung in Device Gruppen durch Verzeichnisse verwirklicht wurden.

```
# Verzeichnis Struktur unter /dev
dev/pts
dev/lat
dev/sad
dev/streams/xtiso
dev/streams
dev/pf
dev

# Auschnitt des /dev Verzeichnis
brw-------   1 root     system     8,2048 Oct 17  1994 rz2a
brw-------   1 root     system     8,2049 Oct 17  1994 rz2b
brw-------   1 root     system     8,2050 Oct 17  1994 rz2c
brw-------   1 root     system     8,2051 Oct 17  1994 rz2d
brw-------   1 root     system     8,2052 Oct 17  1994 rz2e
brw-------   1 root     system     8,2053 Oct 17  1994 rz2f
brw-------   1 root     system     8,2054 Oct 17  1994 rz2g
brw-------   1 root     system     8,2055 Oct 17  1994 rz2h
crw-rw-rw-   1 root     system     9,21511 Nov 14  1995 nrmt0a
crw-rw-rw-   1 root     system     9,21507 Sep  2 13:21 nrmt0h
crw-rw-rw-   1 root     system     9,21505 Nov 14  1995 nrmt0l
crw-rw-rw-   1 root     system     9,21509 Nov 14  1995 nrmt0m
crw-rw-rw-   1 root     system     9,21510 Nov 14  1995 rmt0a
crw-rw-rw-   1 root     system     9,21506 Nov 21 18:42 rmt0h
crw-rw-rw-   1 root     system     9,21504 Nov 14  1995 rmt0l
crw-rw-rw-   1 root     system     9,21508 Nov 14  1995 rmt0m
```

Zum Erzeugen der Special Files, wechselt man ins Verzeichnis /dev und erzeugt dort mit MAKEDEV oder wenn man die *major* und *minor* Device Nummer kennt die nötigen Files.

```
syntax: mknod special_file [ type major# minor ]
```

Wobei `type` angibt ob es sich dabei um ein Character Device (type=c) oder um ein Block Device (type=b) handelt. Für die meisten gängigen Devices gibt es bereits einen Eintrag im Script `MAKEDEV`, was uns das

Suchen nach der *major* und *minor* Device Nummer erspart. Dabei ist zu beachten, dass der Special Filename nicht gleich dem Device Namen ist. So erzeugt man die Special Files zu einem Tape Drive, das am externen SCSI Kontroller hängt durch `MAKEDEV tz13`. Dabei werden die files nrmt0a, nrmt0h, nrmt0l, nrmt0m, rnt0a, rmt0h, rmt0l und rmt0m erzeugt. Die Information wie das Device heisst bekommt man aus der Startup Meldung des Kernels (beim Booten, durch `uerf -R -c oper` oder im File /usr/sys/conf/GENERIC (oder /usr/sys/conf/SYSTEMNAME SYSTEMNAME=Name der Maschine in Grossbuchstaben).

5.2 kernel configuration

OSF/1 besitzt einen statischen Kernel. Sind daher Treiber nicht im aktuellen Kernel enthalten so muss dieser neu konfiguriert und gebildet werden. Dazu wechselt man ins Verzeichnis /usr/sys/conf. Dort befinden sich alle Konfigurationsfiles. Wichtig sind aber nur die Files GENERIC, BINARY und SYSTEMNAME (SYSTEMNAME ist der Name der Maschine in Grossbuchstaben). GENERIC enthält dabei alle Device Treiber Einträge, alle Devices und sonstigen Optionen. Sollte man also nach einer Option suchen, kann man in GENERIC nachschauen. SYSTEMNAME ist das Konfigurations File unserer Maschine. Dieses muss editiert werden. Es enthält meist nur mehr die Device Treiber für jene Geräte, die zur Zeit der Erstinstallation vorhanden waren.

```
ident           "FSTGAL03"

options         GENERIC
options         UERF
options         OSF
options         _LMF_
options         ULT_BIN_COMPAT
options         BIN_COMPAT
options         COMPAT_43
options         MACH
options         MACH_IPC_TCACHE
options         MACH_IPC_WWA
options         MACH_IPC_XXXHACK
options         STREAMS
options         BUFCACHE_STATS
options         INOCACHE_STATS
options         STAT_TIME
options         VAGUE_STATS
options         UFS
options         NFS
options         LDTTY
options         STRKINFO
options         INET
options         UIPC
options         SYSV_COFF
options         SYSV_ELF
options         QUOTA
options         LABELS
options         DLI
options         BSD_TTY
options         LAT
options         DLB
options         MSFS
options         FFM_FS
options PACKETFILTER
options   PCKT
```

```
options         TIRDWR
options         TIMOD
options         XTISO
options         FFM_FS
options         CDFS
options DEC_AUDIT
#
# Standard options.
#
options         UNIX_LOCKS
options         SER_COMPAT
options         RT_PREEMPT
options         RT_SCHED
options         RT_SCHED_RQ
options         RT_PML
options         RT_TIMER
options         RT_SEM
options         RT_CSEM
options         RT_IPC
```

Der erste Teil enthält den Namen des Systems `ident "FSTGAL03"` und die aufgenommenen Systemoptionen wie zum Beispiel UFS für das BSD Filesystem oder NFS für das Netzwerk Filesystem.

```
makeoptions     CDEBUGOPTS="-g3"
makeoptions     PROFOPTS="-DPROFILING -DPROFTYPE=4"
#
# Max number of processors in the system (DO NOT CHANGE)
#
processors      16

#
# Special options (see configuring the kernel chapter
# in the Guide to System Administration)
#
timezone        0 dst 0
dfldsiz         134217728
maxdsiz         1073741824
dflssiz         2097152
maxssiz         33554432
cpu             "DEC3000_500"
maxusers        32
machine         alpha

config          vmunix  swap generic
```

In diesem Teil werden die Optionen fuer den Compiler und den Linker angegeben (makeoptions). Und die Optionen, die das System etwas tiefergehend charakterisieren wie zum Beispiel maxsdsiz und maxssiz, die den maximalen virtuellen Memory für einen Prozess bezüglich dessen DATA- und STACK Bereichs festlegen. Der Parameter maxusers beschränkt allerdings nicht die anzahl der User im System sondern es wird von diesem Wert ausgehend die Grösse einiger Systemtabellen festgelegt.

```
bus             tc0         at nexus?
callout after_c "../bin/mkdata tc"
bus             tcds0       at tc0       slot  6 vector    tcdsintr
controller      scsi0       at tcds0     slot  0
device disk     rz2         at scsi0     drive     16
```

```
device disk     rz3       at scsi0    drive     24
device disk     rz4       at scsi0    drive     32
device disk     rz6       at scsi0    drive     48
controller      scsi1     at tcds0    slot  1
device tape     tz13      at scsi1    drive     104
controller      scc0      at tc0      slot  7 vector    sccintr
controller      bba0      at tc0      slot  7 vector    bbaintr
controller      ln0       at tc0      slot  7 vector    lnintr
controller      fb0       at tc0      slot  0 vector    fbint

scs_sysid       1
pseudo-device   lsm     1
pseudo-device   lsm_ted 0
pseudo-device   sysv_hab
pseudo-device   svid_three_hab
pseudo-device   svr_four_hab
pseudo-device   soe_two_hab
pseudo-device   rt_hab
pseudo-device   rpty    255
pseudo-device   ether
pseudo-device   sl      2
pseudo-device   loop
pseudo-device   strpush 16
pseudo-device   prf    6
#
# Max number of processors in the system (DO NOT CHANGE)
#
pseudo-device   cpus    16
pseudo-device   ws
```

Weiters sind auch noch einige Einträge für das Bus System, die Kontroller und Devices (Platten, Tapes..) vorhanden, sowie Pseudodevices wie ether und loop.

Nachfolgend eine Zusammenstellung der wichtigeren Parameter.

```
keyword     default       Beschreibung
ident       hostname      Dieser Name identifiziert den Systemkern
tz          0 dst 0       Zeitzone
dfldsiz     134217728     Default Data Segment Size Limit
maxdsiz     107374182     Maximum Data Segment Size Limit
dflssiz     1048576       Default Stack Size Limit
maxssiz     33554432      Maximum Stack Size Limit

maxusers    32            Beeinflusst einige Systemgroessen
                          Sollte ueber der Zahl der normalerweise
                          eingeloggten Benutzer liegen

maxthreads  256           Maximale Anzahl der Threads pro Prozess
threadmax   8192          Maximale Anzahl der Threads des Systems

maxproc     nproc in      Definiert die maxiamle Anzahl von Prozessen
            param.c       fuer das System
maxuprc     64            Maximale Anzahl der Prozess pro Benutzer
maxvas      1073741824    Maximaler virtueller Adressraum fuer eine user map
swapbuffers 128           Maximale Anzahl von Buffern f\"ur Swap I/O
```

```
MAX_BDEVSW  70
MAX_CDEVSW  125           legen die Groesse der Switch Tabelle fuer die
                          Block und Character Devices fest

CDFS                      ISO 9660 CDROM Unterstuetzung
COMPAT_43                 Rueckwertskompatibilitaet mit 4.3 BSD Filesystem
QUOTA                     UFS Disk Quotas
MSFS                      Advanced Filesystem
NFS                       Network Filesystem
UFS                       Unix Filesystem

OSF                       OSF/1 Systemkern
GENERIC                   generischer Systemkern
MACH                      Standard Mach Eigenschaften
MACH_NET                  schneller Netzzugriff
ULT_BIN_CO                Ultrix Binaerkompatibilitaet

AUTONICE                  Automatisches reduzieren der Prioritaet von Jobs,
                          die die CPU laenger als 10 min in Anspruch nehmen um
                          einen Wert von 4.
```

Hat man das Konfigurationsfile fertig wechselt man in den Single User Mode, mountet das /usr Verzeichnis und führt in /usr/sys/conf den Befehl `doconfig -c SYSTEMNAME` auf. Der Output des Befehls `doconfig` gibt Aufschluss darauf wo der neue Kernel zu finden ist. Sicherheitshalber sichert man den alten Kernel und speichert den neuen als /vmunix. Danach Rebootet man.

6 Druckermanagment

Der allgemeine Vorgang beim Drucken von OSF/1 aus ist folgender: Der Benutzer verwendet `lpr` um sein File an den Drucker zu senden. Das File wird zuerst in das Drucker Spool Verzeichnis gelegt und der Drucker Daemon informiert, dass ein File zur Bearbeitung vorliegt. Die weitere Kontrolle fällt an den Druckerdaemon (`lpd`). Dieser arbeitet den Spool Bereich ab und reicht die Aufträge an den Drucker weiter. Die Konfiguration der Drucker finden die Programme im /etc/printcap File.

6.1 printcap

In /etc/printcap ist jeder Drucker definiert, wobei folgende Parameter möglich sind:

Name	Type	Default	Beschreibung
af	string	NULL	Name des Accounting File
br	nummer	none	Baud Rate, falls Serieller Drucker
df	string	NULL	Der tex Data Filter (DVI format)
ic	bool	false	Driver unterstuetzt ioctl
if	string	NULL	Text Filter der das Accounting uebernmimmt
lf	string	/dev/console	Error Logfile
lo	string	lock	Lock File
lp	string	/dev/lp	Device Name
mx	nummer	1000	Maximale File Groesse
of	string	NULL	Output Filter
pl	nummer	66	Seiten Laenge
pw	nummer	132	Seiten Breite
px	nummer	0	Seiten Breite in Pixel
py	nummer	0	Seiten Laenge in Pixel

rm	string	NULL	Hostname fuer Remote Print Access
rp	string	lp	Name des Remote Printers
rs	bool	false	Beschraenke Remote User
sb	bool	false	Short Banner
sc	bool	false	Unterdruecke Mehrfach Kopien
sd	string	/usr/spool/lpd	Spool Verzeichnis
sh	bool	false	Unterdruecke Burst Page Header
st	string	status	Status Filename
tr	string	NULL	Trailer String, fuer Queue is empty
xc	nummer	0	clear local mode bits
xf	string	NULL	Pass Through Filter

Die Schnittstelle zum lokalen Drucker, der an der parallelen Schnittstelle des Rechners angeschlossen werden kann, heisst /dev/lp0. Ein typischer `printcap` Eintrag würde nun wie folgt aussehen:

```
Drucker|Beschreibender Name:\
        :lp=/dev/lp0:\                  # Device
        :sd=/var/spool/lpd/Drucker:\    # Spool Dir.
        :lf=/var/log/Drucker.log:       # Log File
```

Für einen Drucker, der an einem anderen Rechner hängt und der über das Netzwerk angesprochen werden kann, gibt es die Einträge `rm` für *Remote Machine* und `rp` für *Remote Printer*. *Remote Printer* steht dabei für den Namen unter dem der Drucker an der *Remote Machine* bekannt ist. Ein typischer *Remote Printer* Eintrag schaut wie folgt aus.

```
Drucker|Beschreibender Name:\
        :lp=:\                          # Device loeschen
        :sd=/var/spool/lpd/Drucker:\    # Spool Dir.
        :lf=/var/log/Drucker.log:\      # Log File
        :rm=Hostname:\
        :rp=Druckername:
```

Beim Drucken über das Netz muss noch darauf geachtet werden, dass die *Remote Machine* das Drucken auch zulässt. Dazu kann man im /etc/host.lpd die Rechner mit ihrem Hostnamen eintragen, denen das Drucken erlaubt sein soll.

Zur Druckersteuerung haben wir die Programme

lpr

Drucken eines Dokuments

lprm

Löschen eines Druckjobs; als root kann man alle Aufträge löeschen - als Benutzer nur die eigenen.

lpq

Druckerqueue anzeigen

lpc

Starten, stopen, löschen einer Druckerqueue

7 User Managment

7.1 Das Passwort File

In ihm sind alle Benutzer des Systems aufgelistet.

```
root:c9k5nx0zCNREc:0:1:system PRIVILEGED account:/:/bin/tcsh
nobody:*Nologin:65534:65534:anonymous NFS user:/:
nobodyV:*Nologin:60001:60001:anonymous SystemV.4 NFS user:/:
daemon:*:1:1:system background account:/:
bin:*:3:4:system librarian account:/bin:
uucp:Nologin:4:2:UNIX-to-UNIX Copy:/usr/spool/uucppublic:/usr/lib/uucp/uucico
uucpa:Nologin:4:2:uucp adminstrative account:/usr/lib/uucp:
auth:*:6:11:Authentication Subsystem:/tcb/bin:
cron:*:7:14:Cron Subsystem:/usr/adm/cron:
lp:*:8:12:Line Printer Subsystem:/users/lp:
tcb:*:9:18:Trusted Computing Base:/tcb:
adm:*:10:19:Administration Subsystem:/usr/adm:
blasch:/4gM92YqLM6gI:701:15:Mr.Blasch:/al03_1/users/blasch:/bin/csh
```

Jeder User ist in einer Zeile voll definiert. Der Aufbau der Zeile schaut wie folgt aus:

```
Login Name:Passwort:UID:GID:Name und sonstige Beschreibung:Home:Login Shell
```

Loginname:

Der Loginname muss eindeutig sein. Er wird beim Einloggen als auch bei der Anzeige von Datei zugehörigkeit und vielen mehr verwendet. Kommt ein Loginname mehrmals vor, so wird nur der erste Eintrag verwendet. Einige Benutzernamen werden vom System benutzt.

root

Der privilegierte Account. Von ihm aus kann jeder Befehl ausgeführt werden

nobody

Wird verwendet um Prozesse und Filezugehörigkeit bei NFS und TFTP abzubilden.

daemon

Die meisten Dämonen laufen unter diesem Accountnamen

bin

Der Inhaber der Systembibliotheken und -programme

Passwort

In diesem Feld steht das verschlüsselte Passwort sofern man nicht das Authorisation System von OSF/1 verwendet. Dieses legt die Passwörter und zusätzliche Informationen zum User in das Verzeichnis /tcb/auth/files . mit `edauth loginname` kann dieses editiert werden.

UID und GID

Sind User ID und Group ID zum User. Die User ID muss eindeutig sein.

Name und Beschreibung

Hier können der volle Name und eine persönliche Beschreibung zum Benutzer wie Telefonnummer und Addresse abgelegt werden.

Login Directory (Home)

Das login directory ist jenes Verzeichnis in das der Benutzer gleich nach dem Einloggen gelangt. Ist das Verzeichnis nicht vorhanden oder sind die Filepermissions falsch gesetzt kann es passieren, dass das System das Einloggen des Users völlig unterbindet.

Login Shell

Dies ist die Shell die beim Einloggen gestartet wird. Sie kann vom User mit chsh verändert werden. Es kann auch ein anderes Programm hier verwendet werden. Will man zum Beispiel ermöglichen dass auch andere User einen 'halt' der Maschine durchführen können so kann hier /sbin/halt als login shell eingetragen werden. Will man einen User ganz aus dem System sperren kann hier /sbin/false eingetragen werden. Wenn man Shells verwendet, die nicht zum Systempaket gehören sollte man sie in das File /etc/shells eintragen. Dies ist ein häufig vorkommender Fehler, er zeigt sich meist dadurch, dass sich die Benutzer zwar mit 'telnet' einloggen können nicht aber mit 'ftp'.

7.2 Das Group File

Um das Zugreiffen auf bestimmte Informationen, Programme und Daten auf dem Rechner zu staffeln kann man User in verschiedene Gruppen unterteilen. Die Gruppen werden dazu im File /etc/group aufgelistet.

```
system:*:0:root,blasch,rappel,edvz
daemon:*:1:daemon
uucp:*:2:uucp
mem:*:3:
kmem:*:3:root
bin:*:4:bin,adm
sec:*:5:
mail:*:6:mail
terminal:*:7:
tty:*:7:root
news:*:8:uucp
opr:*:9:root
auth:*:11:
lp:*:12:
lpr:*:12:root
backup:*:13:
cron:*:14:
users:*:15:rappel,klauser,karisch,rathmayr,koegler
sysadmin:*:16:
tape:*:17:
tcb:*:18:
adm:*:19:adm
operator:*:20:
ris:*:21:
edvz:*:700:blasch,edvz,chmr,klauser,nico
ptchemie:*:535:ramek,kelterer
math:*:501:
```

Wichtig ist hier, dass jene Benutzer, die auch `root` Access mittels dem Kommando `su` erreichen sollen, in der Gruppe `system` aufscheinen müssen. Der Gruppen Eintrag ist wieder in mehrere Felder aufgeteilt:

```
Gruppen Name:Passwort:GID:Mitglieder der Gruppe
```

Ein Benutzer kann in eine andere Gruppe mittels `newgrp` wechseln. Aber nur dann wenn er in dieser Gruppe einen Eintrag im `group` File hat.

7.3 Einrichten von Accounts

Auf OSF/1 erfolgt das Einrichten von Accounts mit dem Befehl `adduser`.

7.4 Message of the Day

Das File /etc/motd kann verwendet werden um Benutzern Hinweise auf Systemstörungen, neue Software und sonstige Änderungen zu geben. Dazu kann einfach das File /etc/motd editieren.

7.5 Zusätzliche Login Scripts

Beim Einloggen in das System werden von der Login Shell mehrere Scripts gestartet, die eine Anpassung der Systemvariablen an lokale Gegebenheiten zulassen.

profile

Das File /etc/profile wird von der Shell sh verwendet.

cshrc

Die csh liest als erstes das File /etc/cshrc

.profile

Will der Benutzer eigene Variablen setzen so kann er das in seinem Home Verzeichnis im File .profile erledigen. Dieses wird von der Shell sh gelesen.

.cshrc

Wie .profile nur für die Shell csh.

.login und .logout

Diese beiden Files werden von der Shell csh gelesen wenn man sich in das System das erste mal einloggt bzw. wenn man sich ausloggt.

8 Processkontrolle

Wird auf einem Unix Rechner ein Programm gestartet bekommt es eine eindeutige Prozess ID (PID) zugeordnet. An diese kann man Signale schicken. Angezeigt können die laufenden Prozesse mit dem Kommando `ps` werden.

8.1 Der Befehl ps

Der Befehl `ps` gibt uns eine Liste aller im System befindlichen Prozesse. Der Befehl `ps` kann mit folgenden Optionen aufgerufen werden:

```
syntax: ps [aexgmlvusjrSUALwhT] [Oo format] [ttty] [process number]
Option      Beschreibung
  a   Asks for information regarding processes associated with terminals
      (ordinarily only one's own processes are displayed).

  A   Increases the argument space.

  e   Asks for the environment to be printed, as well as the arguments to the
      command.

  g   Asks for all processes.  Without this flag, ps only prints interesting
      processes.  Processes are deemed to be uninteresting if they are pro-
```

```
    cess group leaders.  This normally eliminates top-level command inter-
    preters and processes waiting for users to log in on free terminals.

h   Repeats the header after each screenful of information.

j   Produces job control information, with fields specified by user, ppid,
    pgid, sess, and jobc.

l   Asks for a detailed list, with fields specified by ppid, cp, pri, nice,
    vsize, rssize and wchan.

L   Lists all available format specifiers.

m   Prints all threads in a task, if the task has more than one.

o specifier[=header],...
    Specifies a list of format specifiers to describe the output format.

O specifier[=header],...
    Same as o, except it displays the fields specified by pid, state,
    tname, cputime, and comm in addition to the specifiers supplied on the
    command line.

s   Gives signal states of the processes, with fields specified by uid,
    cursig, sig, sigmask, sigignore, and sigcatch.

S   Prints usage summaries (total usage of a command, as opposed to current
    usage).

ttty Lists only processes for the specified tty.

T   Lists all processes on your tty.

u   Produces a user oriented output. This includes fields specified by
    user, pcpu, pmem, vsize, rssize, and start.

v   Produces a version of the output containing virtual memory statistics.
    This includes fields specified by cputime, sl, pagein, vsize, rssize,
    pcpu, and pmem.

w   Uses a wide output format (132 columns (bytes) rather than 80); if this
    flag is doubled (ww), uses an arbitrarily wide output.  This informa-
    tion determines how much of long commands to print.

x   Asks even about processes with no terminal.

process_number
    Restricts output to the specified process.  This argument must be
    entered last on the command line.
```

```
# ps axu
USER       PID %CPU %MEM   VSZ  RSS TTY   S    STARTED  TIME     COMMAND
root         9  0.0  0.3 1.04M 816K ??    I      Oct 21  0:00.05 kloadsrv
root      8202  0.0  0.1 1.48M 264K ??    I      Dec 05  0:00.03 rpc.pcnfsd
root         0  0.0  7.7  510M  200 ??    R <    Oct 21 54:09.68 [kernel idle]
root        37  0.0  0.0  184K  32K ??    S      Oct 21 12:28.75 /sbin/update
```

```
root       139  0.0  0.1 1.34M 152K ??    I      Oct 21  0:00.48 binlogd
root       137  0.0  0.1 1.36M 136K ??    S      Oct 21  0:15.88 syslogd
root     16538  0.0  0.1  680K 216K ttyp1 R  + 07:48:24  0:00.02 ps uaxww
root     31899  0.0  0.1 1.55M 312K ??    I      Dec 05  0:00.05 mountd -i -n
```

Bei der Ausgabe werden verschiedene Informationen ausgegeben:

```
Feld     Beschreibung

USER     Benutzername oder UID
UID      User ID
PID      Prozess ID
PPID     Parent Prozess IP, PID des Prozesses der den Prozess gestartet hat
%CPU     Anteil an der verbrauchten CPU Zeit
%MEM     Anteil an der Memory Nutzung
CP       Kurzzeit CPU Auslastung (wird zum Scheduling verwendet)
PRI      Scheduling Prioritaet
NI       Nice Value
VSZ      Virtuelle Groesse des Prozesses in Kilobytes
RSS      Tastaechliche Groesse des Anteils im Hauptspeicher in 1KB Pages
WCHAN    Adresse eines Ereignisses, auf das der Prozess wartet
TTY      Name des Terminals, von dem aus der Prozess gestartet wurde.
S        Prozess Status
STARTED  Start Zeitpunkt
TIME     Bisher benoetigte CPU Zeit
COMMAND  Befehl der zum Starten des Prozesses verwendet wurde
```

Der Prozess kann folgenden Status annehmen

```
Status   Beschreibung

R        Prozess laeuft, oder ist lauffaehig
U        Prozess kann nicht unterbrochen werden und schlaeft
S        Prozess schlaeft seit weniger als 20 Sekunden
I        Prozess ist idle (schl\"aft bereits laenger als 20 Sekunden)
T        Prozess ist gestopt
H        Prozess ist angehalten
W        Prozess ist auf das Swap Device ausgelagert
>        der Prozess hat ein Soft Limit gesetzt und ueberschreitet dieses
N        Prozess Prioritaet ist reduziert
<        Prozess Prioritaet wurde kuenstlich erhoeht
+        Prozess ist ein Eltern Prozess mit einem kontrollierenden tty
```

8.2 Anhalten und Beenden von Prozessen

Wie am Anfang schon erwähnt kann man den Prozessen über deren ProzessID Signale schicken. Der Befehl dazu ist `kill -SIGNAL PID` . Die wichtigsten Signale sind

```
SIGNAL NAME    Beschreibung

1      HUP     Hang up, bewirkt bei den meisten Daemonen einen Restart
               des Services und damit ein neu einlesen der Konfigurations
               Files
3      QUIT    Quit
9      KILL    Abbrechen des Prozesses
15     TERM    Software Terminierungssignal
```

```
17      STOP    Haelt den Prozess an
19      CONT    mit diesem Signal wird ein mit STOP angehaltener Prozess
                fortgesetzt
```

8.3 Verändern der Priorität von Prozessen

Auch die Prozesspriorität kann nachträglich geändert werden. Der Befehl dazu heisst `renice`. Mit der Priorität kann man die Zuordnung von CPU Zeit verändern.

```
syntax: renice -n priority [-p] [-g | -u] ID ...
        renice priority [-p] pid ...  [-g pgrp ...] [-u user ...]
```

Die möglichen Werte liegen zwischen der höchsten Priorität -20 und der niedrigsten 20, wobei Werte von -20 bis -1 nur vom 'root' vergeben werden können. 0 ist gleichbedeutend dem interaktiven Rechnen, und eine Priorität von 20 teilt dem Prozess nur dann Rechenzeit zu wenn kein anderer Prozess läuft. Verwendet man die Syntax mit dem Flag -n so wird der Wert zum momentanen dazuaddiert. Mit dem Flag -g wird die ID als Group ID interpretiert mit -u als User ID. Will man einen Prozess gleich mit einer geänderten Priorität starten so kann man `nice -n command` verwenden.

9 Datensicherheit und Sicherungsstrategien

Für das Backup von Daten stehen uns auf OSF/1 die Befehle dump, restore, dd und tar zur Verfügung.

9.1 Der tar Befehl

Das tar Programm fasst mehrere Files oder Verzeichnisse in ein File zusammen. Zum Zusammenpacken der Programme verwendet man

```
tar cfv Packfile.tar Files ...
tar cfv Device Files ...
```

Das Auspacken erfolgt mit

```
tar xfv Packfile.tar Files ...
tar xfv Device Files ...
```

Den Inhalt kann man mit

```
tar tfv Packfile.tar Files ...
tar tfv Device Files ...
```

auflisten. Wichtig sind auch noch die Flags

Flag s

Löscht führende Slash (/). Sonst wird ein absoluter Pfad angenommen.

Flag p

Restauriert die Originalmodes

9.2 Der dd Befehl

Mit dem Befehl `dd` kann man einen Spiegel eines Devices anlegen. Er kopiert den Datenträger Byte für Byte. Sowohl der Input als auch der Output, die mit `if=` bzw. `of=` angegeben werden, können mit Raw Devices arbeiten.

```
syntax: dd if=inputfile of=outputfile options conv=specs
```

Folgende wichtige Optionen stehen zur Verfügung

ibs=

Gibt die Blockgrösse für das Input Device an

obs=

Gibt die Blockgrösse für das Output Device an

bs=

Bei gleicher Blockgösse für Input und Output kann `bs` verwedet werden.

conv=

Hier können Konvertierungen vorgenommen werden. Einige der Formate sind ascii (ascii -> ebcdic), ebcdic (ebcdic->ascii) und ibm (ein leicht verändertes ebcdic format -¿ ascii).

9.3 dump und restore

Mit dump und restore stehen interaktive Tools zur Verfügung mit denen man Filesysteme sichern kann. Das Sicherungsmedium kann mit `-f device` angegeben werden.

```
dump -f[0-9] device filesystem
restore -if device
```

Bei dump kann man den Dump Level angeben. Dump Level 0 ist dabei ein vollständiges Backup die Level 1 bis 9 sind Incremental Backups wobei der anschliessende Level die Files ins Backup aufnimmt, die sich zum vorangegangenen Dumplevel geändert haben.

Im Programm `restore` hat man die Befehle `cd`, `pwd` und `ls` zur Verfügung um sich in der Filesystemstruktur zu bewegen. Mit `add filename` und `delete filename` kann man Files und Verzeichnisse in die Liste der zu extrahierenden Files aufnehmen und diese schliesslich mit `restore` vom Sicherungsmedium zurückholen. Soll ein ganzes Filesystem zurück geholt werden so wechselt man in das Verzeichnis, in das das Filesystem eingehängt werden soll und verwendet `restore -rf device` um das Filesystem ohne interaktive Abfragen zurückzuholen.

10 TCP/IP

Das TCP/IP besteht aus den Elementen

- Internet Protocol (IP) zum Transport der Daten
- Internet Control Message Protocol (ICMP) Dieses unterstützt IP bei Fehlermeldungen und liefert Routing Information.
- Address Resolution Protocol (ARP) führt die Umsetzung von IP Adressen in Hardwareadressen durch.

- Transport Control Protocol (TCP) und User Datagram Protocol (UDP). Werden zum Datenaustausch zwischen Programmen auf Basis von IP verwendet. Während TCP verbindungsorientiert arbeitet, dass heisst die korrekete Verbindung zwischen Sender und Empfänger sicherstellt, arbeitet UDP ohne die Sicherstellung.

10.1 Der Routing Mechanismus und Domain Name Service

Der Routing-Mechanismus ist notwendig um in einem stark verzweigten Netz den Weg zwischen Sender und Empfänger zu finden. Der Domain Name Service hilft dem Benutzer sich im Netz zurechtzufinden.

Zunächst versucht der Rechner den Namen des Rechners in seine IP Adresse umzusetzen. Dazu schaut er zunächst in das File /etc/svcorder. Dieses enthält die Information in welcher Reihenfolge er den Rechner im lokalen Host-File und/oder am Domain Name Server (DNS) sucht. Hat man keinen DNServer zur Verfügung so kann man den Namen mit der dazugehörigen Adresse in das /etc/hosts File eintragen.

```
# Description:  The hosts file associates hostnames with IP addresses.
#
# Syntax:  nnn.nnn.nnn.nnn hostname.domain.name[alias_1,...,alias_n] [#comments]
#
# nnn.nnn.nnn.nnn       the IP address of the host
# hostname.domain.name  the fully qualified hostname, including the domainname
# alias_n               other names or abbreviations for this host
# #comments             text following the comment character (#) is ignored
#
127.0.0.1 localhost
129.27.9.201 fstgal03.tu-graz.ac.at fstgal03
129.27.179.2 fptchbds01.tu-graz.ac.at fptchbds01
```

Der eigene Hostname muss auf jeden Fall im /etc/hosts eingetragen sein. In welcher Domain er sich selbst befindet holt er sich aus dem File /etc/resolv.conf. In diesem ist auch die Adresse des dazugehörigen DNServer vermerkt.

```
; Description:  The resolv.conf file lists name-value pairs that provide
;               information to the BIND resolver.
;
; Syntax:       domain  <domainname>
;                       and
;               nameserver  <address>
;
; Caution:  White space entered after the domain name is not ignored; it
;           is interpreted as part of the domain name.
;
; domain <domainname>       local domain name
; nameserver <address>      Internet address of a name server that the
;                           resolver should query
;
domain          tu-graz.ac.at
nameserver      129.27.2.3
```

Die Adresse besteht aus vier Bytes. Jedes Byte kann einen Wert zwischen 0 und 255 annehmen. Wobei einige Werte besondere Bedeutung haben. Da nicht alle Rechner an einem Netzwerkstrang hängen hat man das Netz in Subnets unterteilt. Es gibt davon 5 Klassen wobei die drei wichtigeren das A,B und C Subnet sind. Die vier Bytes der Netzwerkadresse werden dabei unterschiedlich dem Netz(N) oder dem Host(H) zugeordnet.

```
Klasse  1.Byte  Adressformat  Anzahl der moeglichen Rechner

A       1-126   N.H.H.H       16.387.064
B       128-191 N.N.H.H       64.516
C       192-223 N.N.N.H       254
```

Die Nummer 127 im ersten Byte hat die Bedeutung, dass es sich um die Loopback-Schnittstelle handelt. Die Nummer 255 kennzeichnet eine Broadcast Adresse. Damit der Rechner weiss in welcher Netzklasse er sich befindet gibt es die Netmask. Sie maskiert den Host Anteil aus. So gilt für ein Klass C Netz die Maske 255.255.255.0. Die Broadcast Adresse schliesst alle Rechner in einem Subnet ein. Für ein Klass C Netz ist sie N.N.N.255. Die TU-Graz hat gesamt gesehen ein Klass-B Netz mit der Netz Nummer 129.27.H.H jedem Institut wurde jedoch ein oder mehrere Klass-C Netze übergeben. Jede IP Adresse hat auch eine eindeutige Hardwareadresse. Während die IP Adresse in nicht zusammenhängenden Netzwerken mehrmals vorkommen können ist die Hardwareadresse auf der ganzen Welt eindeutig definiert. Sie ist auf der Interface Karte fixiert. Die Zuordung Hardware <-> IP Adresse verwaltet der Rechner in der ARP-Tabelle.

Über die Netzmaske erkennt der Rechner ob sich der gesuchte Host im lokalen Netzwerk befindet oder ob er, um den Host zu erreichen, einen Router braucht. Der Router vermittelt die Pakete zwischen den einzelnen Subnets und ist in /etc/routes eingetragen. Ihm schickt der Rechner alle Pakete zu Rechnern, die er nicht im eigenen Subnet findet.

10.2 Konfiguration und Überprüfung

Konfigurieren kann man die meisten Parameter mit dem setup Programm. Nur selten sollte man die Konfigurationsfiles selber editieren müssen. Die meisten Parameter des Systems kann man mit dem Befehl `netstat` überprüfen. Die Erreichbarkeit von Rechnern mit `ping`. Will man die Route Tabelle überprüfen kann man das mit `netstat -r` erledigen.

```
syntax:   netstat [-Aan] [-f address_family] [-p protocol] [interval]
          netstat [-adHimMnrstu] [-f address_family] [-p protocol] [interval]
          netstat [-ntd] [-I interface] [interval]

Flags:

  -a  Displays the state of sockets related to the Internet protocol.
      Includes sockets for processes such as servers that are currently
      listening at a socket but are otherwise inactive.

  -A  Displays the address of any protocol control blocks associated with
      sockets.  Typically, this flag is used for debugging.

  -d  Displays the number of dropped packets; for use with the -I interface
      or -i flags.  You can also specify an interval argument (in seconds).

  -f address_family
      Limits reports to the specified address family.  The address families
      that can be specified might include the following:

      inet    Specifies reports of the AF_INET family, if present in the ker-
              nel.

      unix    Specifies reports of the AF_UNIX family, if present in the ker-
              nel.
```

```
    ns      Specifies reports of the AF_NS, if present in the kernel.

    all     Lists information about all address families in the system.

    any     Lists information about any address families in the system.

-H  Displays the current ARP table (behaves like arp -a).

-i  Displays the state of configured interfaces.  (Interfaces that are
    statically configured into the system, but not located at system start,
    are not shown.)

    When used with the -a flag, it displays IP and link-level addresses
    associated with the interfaces.

    You can use the -i flag to retrieve your system's hardware address.

-I interface
    Displays information about the specified interface.

-I interface -s
    Displays the DNA Data Link Layer counters for the network interface and
    the adapter's status and characteristics.

-m  Displays information about memory allocated to data structures associ-
    ated with network operations.

-M  Displays Internet protocol multicast routing information. When used
    with the -s flag, it displays IP multicast statistics.

-n  Displays network address in numerical format.  When this flag is not
    specified, the address is displayed as hostname and port number.  This
    flag can be used with any of the display formats.

-pprotocol
    Displays statistics for protocol, which you can specify as a well known
    name or an alias.  Supported protocol names and their aliases are
    listed in /etc/protocols.  A null listing (0) means that there is no
    data to report.  If routines to report statistics for a specified pro-
    tocol are not implemented on this system, netstat reports that the pro-
    tocol is unknown.

-r  Displays the host's routing tables.  When used with the -s flag, shows
    the host's routing statistics instead of routing tables.

-s  Displays statistics for each protocol.

-t  Displays timer information; for use with the -I[interface] or -i flags.

-u  Displays information about domain sockets (UNIX domain).
```

10.3 Das File /etc/services

Es dient lediglich dazu die Port Nummern, die zu den Services gehören zu benennen. Da man nicht für jeden Dienst eine Netzwerkschnittstelle und IP Nummer reservieren kann gibt es Ports die eine weitere Differenzierung der Netzdienste vornehmen. Dabei sind die Ports 0-1023 privilegierte Ports und können nur vom 'root' benutzt werden.

```
Ausschnitt:
# Syntax:  ServiceName PortNumber/ProtocolName [alias_1,...,alias_n] [#comments]
#
# ServiceName           official Internet service name
# PortNumber            the socket port number used for the service
# ProtocolName          the transport protocol used for the service
# alias                 unofficial service names
# #comments             text following the comment character (#) is ignored
#
echo            7/tcp
echo            7/udp
discard         9/tcp           sink null
discard         9/udp           sink null
systat          11/udp          users
daytime         13/tcp
daytime         13/udp
netstat         15/tcp
quote           17/udp          text
chargen         19/tcp          ttytst
chargen         19/udp          ttytst
ftp             21/tcp
telnet          23/tcp
smtp            25/tcp          mail
time            37/tcp          timserver
time            37/udp          timserver
name            42/tcp          nameserver
whois           43/tcp          nicname
domain          53/udp          nameserver      # domain name server
domain          53/tcp          nameserver
```

10.4 Der inetd und seine Konfigurationsdatei inetd.conf

Per Default wird beim Booten der Internet Daemon (inetd) geladen. Er lauscht am Netz ob ein Packet an einem Port anliegt und startet dann den dazugehörigen Service Daemon. In seinem Konfigurationsdatei sind die Service Daemons den Ports und dem Protokoll zugeordnet.

```
# Internet server configuration database
#
# Description:  The inetd.conf file is the file that the inetd daemon reads
#               for information on how to handle Internet service requests.
#
# Syntax:  ServiceName  SocketType  ProtocolName  Wait/NoWait  UserName \
#           ServerPath  ServerArgs
#
# ServiceName     name of an Internet service defined in the /etc/services file
# SockettType     type of socket used by the service, either stream or dgram
# ProtocolName    name of an internet protocol defined in the /etc/protocols
#                 file
```

```
# Wait/NoWait     determines whether the inetd daemon waits for
#                 a datagram server to release the socket before continuing
#                 to listen at the socket
# UserName        the login that inetd should use to start the server
# ServerPath      full pathname of the server
# ServerArgs      optional command line arguments that inetd should use to
#                 execute the server
#
ftp     stream  tcp     nowait  root    /usr/sbin/ftpd          ftpd
telnet  stream  tcp     nowait  root    /usr/sbin/telnetd       telnetd
shell   stream  tcp     nowait  root    /usr/sbin/rshd          rshd
login   stream  tcp     nowait  root    /usr/sbin/rlogind       rlogind
exec    stream  tcp     nowait  root    /usr/sbin/rexecd        rexecd
# Run as user "uucp" if you don't want uucpd's wtmp entries.
#uucp   stream  tcp     nowait  root    /usr/sbin/uucpd         uucpd
finger  stream  tcp     nowait  root    /usr/sbin/fingerd       fingerd
#tftp   dgram   udp     wait    root    /usr/sbin/tftpd         tftpd /tmp
comsat  dgram   udp     wait    root    /usr/sbin/comsat        comsat
```

11 Literaturhinweise

1. UNIX System V.4, Jürgen Gublins und Karl Obermayr, 4 Auflage 1995, Springer
2. UNIX System V, A practical guide, Mark G. Sobell, 3. Auflage 1995, Benj. Cummings.
3. UNIX - Wie funktioniert das Betriebssystem? Maurice J. Bach, 1991, Hanser Verlag
4. UNIX System Administation Handbook, Alleen Frisch, 2 Auflage 1995, O'Reilly
5. TCP/IP illustrated, W.Richards Stevens, 1. Auflage 1995, Addison Wesley
6. UNIX - Systemverwaltung Grundlagen, Dr.Peter Dieterich, Regionales Rechenzentrum für Niedersachsen.
7. man pages, rfc's