

Windows IFS (1)

- Ähnliches Konzept wie unter Linux
- IFS-Treiber für verschiedene konkrete Dateisysteme
- Eine Schicht höher: generische Betriebssystemfunktionen für Zugriffe auf Dateisystem

Windows IFS (2)

- Standard-IFS-Treiber
 - NTFS (New Technology File System)
 - FAT (DOS File Allocation Table; FAT12, -16, -32)
 - CDFS (mit „Joliet“-Erweiterung) und UDF
 - ehemals auch HPFS (OS/2 High Performance FS)
- IFS-Treiber von anderen Anbietern
 - ext2ifs (Linux ext2 / ext3)
 - ReiserDriver (Linux ReiserFS)
 - HFS IFS (Apple HFS)
 - „Paragon Alles Mounter“ (ext2/3)

9. Dateisysteme (3)

- 9.3 Virtuelle FS
- 9.3.2 Windows IFS
- 9.4 Praxis: Linux

/home/esser/Daten/Dozent/Folien/bs2-esser-09.odp

9.3.2 Windows IFS

Windows IFS (3)

- **Filesystem Filter Driver**
 - optionale Zusatztreiber, die es erlauben, das Verhalten eines Dateisystems zu verändern
 - mögliche Anwendungen:
 - On-the-fly-Virentfilter
 - Transparente Verschlüsselung
 - Backup-Mechanismen
 - Überwachung
 - Beispiel: Sysinternals FileMon für Überwachung der Dateisystemzugriffe

Windows-Standardfunktionen (2)

- **SetFilePointer (handle, offset, 0, method)**
an bestimmte Stelle in der Datei springen
(abhängig vom letzten Parameter: ab Dateianfang, Dateiende oder ab aktueller Position)
- **GetFileAttributes (filename)**
Dateiattribute abfragen
- **LockFile (handle, offset, length)**
Blöcke in der Datei gegen parallelen Zugriff sperren
- **UnlockFile (handle, offset, length)**
Sperrung aufheben

Windows-Standardfunktionen (1)

- **handle = CreateFile (filename, access, ...)**
Datei erzeugen oder vorhandene Datei öffnen
- **DeleteFile (filename)**
Datei löschen
- **CloseHandle (handle)**
Datei schließen
- **ReadFile (handle, buffer, len, &count, NULL)**
aus geöffneter Datei lesen
- **WriteFile (handle, buffer, len, &count, NULL)**
in geöffnete Datei schreiben

Windows-Standardfunktionen (3)

- **CreateDirectory ()** neues Verzeichnis erzeugen
- **RemoveDirectory ()** leeres Verzeichnis löschen
- **FindFirstFile ()** ersten Eintrag in Verzeichnis lesen
- **FindNextFile ()** nächsten Eintrag lesen
- **MoveFile ()** Datei in anderes Verzeichnis verschieben
- **SetCurrentDirectory ()** Arbeitsverzeichnis wechseln

Windows-Dateiattribute (1)

- **FILE_ATTRIBUTE_ARCHIVE**
Datei soll archiviert werden
- **FILE_ATTRIBUTE_ENCRYPTED**: Datei ist verschlüsselt
- **FILE_ATTRIBUTE_HIDDEN**
Datei nicht in Standard-Listings anzeigen
- **FILE_ATTRIBUTE_OFFLINE**
Datei liegt in Offline-Speicher (z.B. Magnetband)
- **FILE_ATTRIBUTE_READONLY**: nur-lesbar
- **FILE_ATTRIBUTE_SYSTEM**: Systemdatei
- **FILE_ATTRIBUTE_TEMPORARY**: temporäre Datei
- **FILE_ATTRIBUTE_NORMAL**: keine Attribute

```
Sep 19 14:27:41 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 01:00:01 amd64 /usr/sbin/cron[29278]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 20 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 02:00:01 amd64 /usr/sbin/cron[30103]: (root) CMD (/sbin/evlogmgr -c "age > *30d*")
Sep 20 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 12:46:44 amd64 sshd[6516]: Accepted rsa for esser from :ffff:87.234.201.207 port 62004
Sep 20 12:46:44 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 12:48:41 amd64 sshd[6609]: Accepted rsa for esser from :ffff:87.234.201.207 port 62105
Sep 20 12:54:44 amd64 sshd[6694]: Accepted rsa for esser from :ffff:87.234.201.207 port 62514
Sep 20 15:27:35 amd64 sshd[9077]: Accepted rsa for esser from :ffff:87.234.201.207 port 64242
Sep 20 15:27:35 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 16:37:11 amd64 sshd[10102]: Accepted rsa for esser from :ffff:87.234.201.207 port 63375
Sep 20 16:37:11 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 16:38:10 amd64 sshd[10140]: Accepted rsa for esser from :ffff:87.234.201.207 port 63546
Sep 21 01:00:01 amd64 /usr/sbin/cron[17055]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 21 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 21 02:00:01 amd64 /usr/sbin/cron[17878]: (root) CMD (/sbin/evlogmgr -c "age > *30d*")
Sep 21 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 21 17:43:26 amd64 sshd[31088]: Accepted rsa for esser from :ffff:87.234.201.207 port 63397
Sep 21 17:43:26 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 21 17:53:39 amd64 sshd[31269]: Accepted rsa for esser from :ffff:87.234.201.207 port 64391
Sep 21 18:43:26 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 21 19:43:26 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 22 01:00:01 amd64 /usr/sbin/cron[4674]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 22 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 22 02:00:01 amd64 /usr/sbin/cron[5499]: (root) CMD (/sbin/evlogmgr -c "age > *30d*")
Sep 22 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 22 02:00:01 amd64 /usr/sbin/cron[24739]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 23 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 23 02:00:01 amd64 /usr/sbin/cron[25555]: (root) CMD (/sbin/evlogmgr -c "age > *30d*")
Sep 23 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 23 18:04:05 amd64 sshd[6554]: Accepted publickey for esser from :ffff:192.168.1.5 port 6093
Sep 23 18:04:05 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 23 18:04:34 amd64 sshd[6606]: Accepted rsa for esser from :ffff:87.234.201.207 port 62093
Sep 24 01:00:01 amd64 /usr/sbin/cron[12436]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 24 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 02:00:01 amd64 /usr/sbin/cron[13253]: (root) CMD (/sbin/evlogmgr -c "age > *30d*")
Sep 24 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 11:15:48 amd64 sshd[20998]: Accepted rsa for esser from :ffff:87.234.201.207 port 64456
Sep 24 13:49:08 amd64 sshd[23197]: Accepted rsa for esser from :ffff:87.234.201.207 port 61330
Sep 24 13:49:08 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 15:42:07 amd64 kernel: amd_seq_midi_event: unsupported module, tainting kernel.
Sep 24 15:42:07 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 15:42:07 amd64 kernel: amd_seq_osa: unsupported module, tainting kernel.
Sep 24 20:25:31 amd64 sshd[29399]: Accepted rsa for esser from :ffff:87.234.201.207 port 62566
Sep 24 20:25:31 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 01:00:02 amd64 /usr/sbin/cron[662]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 25 01:00:02 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 02:00:01 amd64 /usr/sbin/cron[1484]: (root) CMD (/sbin/evlogmgr -c "age > *30d*")
Sep 25 02:00:02 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 10:59:25 amd64 sshd[8889]: Accepted rsa for esser from :ffff:87.234.201.207 port 64183
Sep 25 10:59:25 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 10:59:47 amd64 sshd[8921]: Accepted rsa for esser from :ffff:87.234.201.207 port 64253
Sep 25 11:30:02 amd64 sshd[9372]: Accepted rsa for esser from :ffff:87.234.201.207 port 62029
Sep 25 11:59:25 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 14:05:37 amd64 sshd[11554]: Accepted rsa for esser from :ffff:87.234.201.207 port 62822
Sep 25 14:05:37 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 14:06:10 amd64 sshd[11866]: Accepted rsa for esser from :ffff:87.234.201.207 port 62951
Sep 25 14:07:17 amd64 sshd[11608]: Accepted rsa for esser from :ffff:87.234.201.207 port 63392
Sep 25 14:08:33 amd64 sshd[11630]: Accepted rsa for esser from :ffff:87.234.201.207 port 63709
Sep 25 15:25:33 amd64 sshd[12930]: Accepted rsa for esser from :ffff:87.234.201.207 port 62778
```

9.4 Praxis: Linux

Windows-Dateiattribute (2)

Erweiterung der klassischen FAT-Dateiattribute:

- archive (A) -> FILE_ATTRIBUTE_ARCHIVE
- read-only (R) -> FILE_ATTRIBUTE_READONLY
- hidden (H) -> FILE_ATTRIBUTE_HIDDEN
- system (S) -> FILE_ATTRIBUTE_SYSTEM

```
Sep 19 14:20:18 amd64 sshd[20494]: Accepted rsa for esser from :ffff:87.234.201.207 port 61557
Sep 19 14:27:41 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 01:00:01 amd64 /usr/sbin/cron[29278]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 20 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 02:00:01 amd64 /usr/sbin/cron[30103]: (root) CMD (/sbin/evlogmgr -c "age > *30d*")
Sep 20 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 12:46:44 amd64 sshd[6516]: Accepted rsa for esser from :ffff:87.234.201.207 port 62004
Sep 20 12:46:44 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 12:48:41 amd64 sshd[6609]: Accepted rsa for esser from :ffff:87.234.201.207 port 62105
Sep 20 12:54:44 amd64 sshd[6694]: Accepted rsa for esser from :ffff:87.234.201.207 port 62514
Sep 20 15:27:35 amd64 sshd[9077]: Accepted rsa for esser from :ffff:87.234.201.207 port 64242
Sep 20 15:27:35 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 16:37:11 amd64 sshd[10102]: Accepted rsa for esser from :ffff:87.234.201.207 port 63375
Sep 20 16:37:11 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 16:38:10 amd64 sshd[10140]: Accepted rsa for esser from :ffff:87.234.201.207 port 63546
Sep 21 01:00:01 amd64 /usr/sbin/cron[17055]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 21 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 21 02:00:01 amd64 /usr/sbin/cron[17878]: (root) CMD (/sbin/evlogmgr -c "age > *30d*")
Sep 21 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 21 17:43:26 amd64 sshd[31088]: Accepted rsa for esser from :ffff:87.234.201.207 port 63397
Sep 21 17:53:39 amd64 sshd[31269]: Accepted rsa for esser from :ffff:87.234.201.207 port 64391
Sep 21 18:43:26 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 21 19:43:26 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 22 01:00:01 amd64 /usr/sbin/cron[4674]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 22 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 22 02:00:01 amd64 /usr/sbin/cron[5499]: (root) CMD (/sbin/evlogmgr -c "age > *30d*")
Sep 22 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 22 02:00:01 amd64 /usr/sbin/cron[24739]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 23 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 23 02:00:01 amd64 /usr/sbin/cron[25555]: (root) CMD (/sbin/evlogmgr -c "age > *30d*")
Sep 23 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 23 18:04:05 amd64 sshd[6554]: Accepted publickey for esser from :ffff:192.168.1.5 port 6093
Sep 23 18:04:05 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 23 18:04:34 amd64 sshd[6606]: Accepted rsa for esser from :ffff:87.234.201.207 port 62093
Sep 24 01:00:01 amd64 /usr/sbin/cron[12436]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 24 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 02:00:01 amd64 /usr/sbin/cron[13253]: (root) CMD (/sbin/evlogmgr -c "age > *30d*")
Sep 24 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 11:15:48 amd64 sshd[20998]: Accepted rsa for esser from :ffff:87.234.201.207 port 64456
Sep 24 13:49:08 amd64 sshd[23197]: Accepted rsa for esser from :ffff:87.234.201.207 port 61330
Sep 24 13:49:08 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 15:42:07 amd64 kernel: amd_seq_midi_event: unsupported module, tainting kernel.
Sep 24 15:42:07 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 15:42:07 amd64 kernel: amd_seq_osa: unsupported module, tainting kernel.
Sep 24 20:25:31 amd64 sshd[29399]: Accepted rsa for esser from :ffff:87.234.201.207 port 62566
Sep 24 20:25:31 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 01:00:02 amd64 /usr/sbin/cron[662]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 25 01:00:02 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 02:00:01 amd64 /usr/sbin/cron[1484]: (root) CMD (/sbin/evlogmgr -c "age > *30d*")
Sep 25 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 10:59:25 amd64 sshd[8889]: Accepted rsa for esser from :ffff:87.234.201.207 port 64183
Sep 25 10:59:25 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 10:59:47 amd64 sshd[8921]: Accepted rsa for esser from :ffff:87.234.201.207 port 64253
Sep 25 11:30:02 amd64 sshd[9372]: Accepted rsa for esser from :ffff:87.234.201.207 port 62029
Sep 25 11:59:25 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 14:05:37 amd64 sshd[11554]: Accepted rsa for esser from :ffff:87.234.201.207 port 62822
Sep 25 14:05:37 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 14:06:10 amd64 sshd[11866]: Accepted rsa for esser from :ffff:87.234.201.207 port 62951
Sep 25 14:07:17 amd64 sshd[11608]: Accepted rsa for esser from :ffff:87.234.201.207 port 63392
Sep 25 14:08:33 amd64 sshd[11630]: Accepted rsa for esser from :ffff:87.234.201.207 port 63709
Sep 25 15:25:33 amd64 sshd[12930]: Accepted rsa for esser from :ffff:87.234.201.207 port 62778
```

9.4.1 Dateizugriff in Linux-Programmen

Dateizugriff in Linux-Programmen (1)

Datei öffnen, lesen, schreiben, schließen (hatten wir schon in *Betriebssysteme I*)

```
fd = open ( "/etc/fstab", O_RDONLY );
while ( (len = read ( fd, line, bufsiz )) > 0 ) {
    if ( len < bufsiz) { line[len]='\0'; }
    printf ("%s", line );
}
close (fd);

fd = open ( "/etc/fstab", O_WRONLY );
write ( fd, buffer, bufsiz );
```

Jetzt mehr Details ...

Dateizugriff in Linux-Programmen (3)

- O_SYNC: Synchroner I/O-Modus (jeden Schreibbefehl sofort ausführen)
- O_NOATIME: Bei Lesezugriffen nicht die Access-Time aktualisieren
- O_LARGEFILE: Datei benötigt 64 Bit zur Größenangabe
- O_NOFOLLOW: Wenn der Dateiname ein symbolischer Link ist, fehlschlagen

- und diverse weitere Flags ...

Dateizugriff in Linux-Programmen (2)

fd = open (filename,flags);

Mögliche Flags:

- O_RDONLY: Nur zum Lesen öffnen
- O_WRONLY: Nur zum Schreiben öffnen
- O_RDWR: Zum Schreiben und Lesen öffnen
- O_CREAT: Datei erzeugen, wenn sie noch nicht existiert
- O_TRUNC: Wenn Datei schon existiert, überschreiben (Länge auf 0 setzen)
- O_APPEND: Append-Modus; Dateizeiger auf Dateiende positionieren

Dateizugriff in Linux-Programmen (4)

int stat (const char *filename, struct stat *buf);

gibt zu einer Datei folgende Eigenschaften zurück:

```
struct stat
{
    dev_t          st_dev;          /* Device (welches Dateisystem?) */
    ino_t          st_ino;          /* INode */
    mode_t         st_mode;         /* Zugriffsrechte */
    nlink_t        st_nlink;       /* Anzahl harter Links */
    uid_t          st_uid;         /* UID des Besitzers */
    gid_t          st_gid;         /* GID des Besitzers */
    dev_t          st_rdev;        /* Gerätetyp (wenn INode-Gerät) */
    off_t          st_size;        /* Größe in Bytes */
    unsigned long  st_blksize;     /* Blockgröße */
    unsigned long  st_blocks;     /* Allozierte Blocks (512-Byte-Blocks) */
    time_t         st_atime;       /* Letzter Zugriff */
    time_t         st_mtime;       /* Letzte Modifikation */
    time_t         st_ctime;       /* Letzte Änderung (von Verwaltungsinf.) */
};
```

Dateizugriff in Linux-Programmen (5)

Beispiel für die Verwendung von stat():

```
/* fileinfo.c */
#include <stdio.h>
#include <sys/stat.h>
#include <stdlib.h>

main () {
    struct stat status;
    int rdev;

    if (stat("/etc/fstab", &status) == -1) {
        return -1;
    } else {
        printf ("Dateigröße: %d \n", status.st_size);
        printf ("UID: %d \n", status.st_uid);
        printf ("GID: %d \n", status.st_gid);
        rdev = status.st_rdev;
        printf ("Geraetedei: (%d,%d) \n", rdev/256, rdev%256);
    }
}
```

Dateizugriff in Linux-Programmen (7)

Zugriffsrechte und Eigenschaften prüfen (1)

S_IFMT	0017000	Bitmaske für die Dateityp-Bitfelder
S_IFSOCK	0140000	Socket
S_IFLNK	0120000	symbolische Verknüpfung
S_IFREG	0100000	reguläre Datei
S_IFBLK	0060000	blockorientiertes Gerät
S_IFDIR	0040000	Verzeichnis
S_IFCHR	0020000	zeichenorientiertes Gerät
S_IFIFO	0010000	FIFO
S_ISUID	0004000	SUID-Bit
S_ISGID	0002000	SGID-Bit (siehe unten)
S_ISVTX	0001000	Sticky-Bit (siehe unten)
S_IRWXU	00700	Bitmaske für Besitzerzugriffsrechte
S_IRUSR	00400	Besitzer hat Lesezugriff
S_IWUSR	00200	Besitzer hat Schreibzugriff
S_IXUSR	00100	Besitzer hat Ausführungsrechte
S_IRWXG	00070	Bitmaske für Gruppenzugriffsrechte
S_IRGRP	00040	Gruppe hat Lesezugriff
S_IWGRP	00020	Gruppe hat Schreibzugriff
S_IXGRP	00010	Gruppe hat Ausführungsrechte
S_IRWXO	00007	Bitmaske für Zugriffsrechte Anderer (nicht in Gruppe)
S_IROTH	00004	Anderer haben Lesezugriff
S_IWOTH	00002	Anderer haben Schreibzugriff
S_IXOTH	00001	Anderer haben Ausführungsrechte

Dateizugriff in Linux-Programmen (6)

```
$ ls -l /etc/fstab
-rw-r--r-- 1 root root 992 2005-04-11 20:24 /etc/fstab
$ fileinfo /etc/fstab
Dateigröße: 992
UID: 0
GID: 0
Geraetedei: (0,0)

$ ls -l /dev/sda3
brw-rw---- 1 root disk 8, 3 2005-03-19 20:36 /dev/sda3
$ fileinfo /dev/sda3
Dateigröße: 0
UID: 0
GID: 6
Geraetedei: (8,3)

$ ls -l /dev/tty22
crw--w---- 1 root tty 4, 22 2005-03-19 20:36 /dev/tty22
$ fileinfo /dev/tty22
Dateigröße: 0
UID: 0
GID: 5
Geraetedei: (4,22)
```

Auszug aus /etc/group:

```
root:x:0:
bin:x:1:daemon
daemon:x:2:
sys:x:3:
tty:x:5:
disk:x:6:
lp:x:7:
www:x:8:
kmem:x:9:
wheel:x:10:
mail:x:12:
```

Erste Zeile aus /etc/passwd:

```
root:x:0:0:
root:/root:
/bin/bash
```

Dateizugriff in Linux-Programmen (8)

Zugriffsrechte und Eigenschaften prüfen (2)

```
struct stat status;
mode_t modus;
stat ("/etc/fstab", &status);
modus = status.st_mode;
if (modus & S_IFREG) { printf ("%s", "Reguläre Datei \n"); }
if (modus & S_IFDIR) { printf ("%s", "Verzeichnis \n"); }
if (modus & S_IFLNK) { printf ("%s", "Symbolischer Link \n"); }

$ testfile /etc/fstab
Reguläre Datei
$ testfile /etc
Verzeichnis
$ testfile /etc/rc.d
Symbolischer Link
$ testfile /dev/zero
$
```

Dateizugriff in Linux-Programmen (9)

Bestimmten Block lesen oder schreiben

Blockgröße: *bsize*, *n*-ter Block (ab 0 zählen)

```
int bsize;
char daten[bsize];

fd = open (filename, O_RDONLY);
lseek (fd, bsize*n, SEEK_SET);
read (fd, &daten, bsize);

fd = open (filename, O_WRONLY);
lseek (fd, bsize*n, SEEK_SET);
write (fd, &daten, bsize);

lseek (fd, bsize*n, SEEK_CUR); /* relativ n Blöcke springen */
```

Dateizugriff in Linux-Programmen (11)

Arbeiten mit Blöcken: Eine Mini-Datenbank

```
struct dbrecord {
    int personid; /* ID für die Person */
    int flag; /* Flag: Dieser Eintrag wird benutzt */
    char nachname[40]; /* Nachname */
    char vorname[40]; /* Vorname */
    char strasse[50]; /* Anschrift */
    char plz[5];
    char ort[40];
};
struct dbrecord dr;
const int reccsize = sizeof (struct dbrecord); /* 184 */

struct dbrecord dummy_record (int i) {
    struct dbrecord d;
    d.personid=i; d.flag=0;
    strcpy (d.nachname, "Mustermann");
    strcpy (d.vorname, "Thomas");
    strcpy (d.strasse, "Lothstrasse");
    strcpy (d.plz, "80000");
    strcpy (d.ort, "Muenchen");
    return d;
}
```

Dateizugriff in Linux-Programmen (10)

Aktuelle Lese-/Schreibposition in Datei bestimmen

```
pos = lseek (fd, 0, SEEK_CUR);
/* springt relativ um 0 Byte, also gar nicht */
/* lseek() liefert immer die neue Position nach dem Sprung zurück */
```

Dateizugriff in Linux-Programmen (12)

```
void init_db () {
    int fd, i;
    fd = open ("/tmp/mytmpfile",
        O_WRONLY|O_CREAT,S_IRREAD|S_IWRITE);
    for (i=0; i<20; i++) {
        dr = dummy_record (i);
        write (fd, &dr, reccsize);
    }
    close (fd);
}

void read_record (int i) {
    struct dbrecord dr;
    int fd;
    fd = open ("/tmp/mytmpfile", O_RDONLY);
    lseek (fd, i*reccsize, SEEK_SET);
    read (fd, &dr, reccsize);
    printf ("ID: %d, Flag: %d \n",
        dr.personid, dr.flag);
    printf ("Nachname: %s \n", dr.nachname);
    printf ("Vorname: %s \n", dr.vorname);
    printf ("Adresse: %s \n", dr.strasse);
    printf ("Ort: %s \n\n", dr.ort);
    close (fd);
    return;
}

main () {
    init_db ();
    read_record (3);
    read_record (11);
}

$ gcc -o mini-db mini-db.c
$ ./mini-db
ID: 3, Flag: 0
Nachname: Mustermann
Vorname: Thomas
Adresse: Lothstrasse
Ort: Muenchen

ID: 11, Flag: 0
Nachname: Mustermann
Vorname: Thomas
Adresse: Lothstrasse
Ort: Muenchen

$ _
```


Dateizugriff in Linux-Programmen (13)

```
$ hexdump /tmp/mytmpfile
00000000 00 00 00 00 00 00 00 00 4d 75 73 74 65 72 6d 61 |.....Musterma
00000010 6e 6e 00 00 00 00 00 00 1c ef ff bf be 66 07 00 |nn.....iÿçÿf..
00000020 60 ef ff bf 18 75 01 40 f2 82 04 08 20 00 00 00 |`iÿç.u.@ð...
00000030 54 68 6f 6d 61 73 00 00 08 db 04 40 d0 7c 04 40 |Thomas...Û.@Þ|.
00000040 a0 7c 01 40 a0 7c 01 40 03 00 00 00 58 7f 01 40 | |.@ |.@...X..@
00000050 88 7f 01 40 f4 81 04 08 4c 6f 74 68 73 74 72 61 |...@...Lothstra
00000060 73 73 65 00 c4 76 01 40 00 00 00 00 d0 ef ff bf |sse.Äv@...Ðiÿç
00000070 9c 82 00 40 f4 6f 01 40 01 00 00 00 18 75 01 40 |...@ð@...u.@
00000080 70 ef ff bf 56 d0 00 40 c4 76 38 30 30 30 30 4d |piÿçVð.@Äv80000M
00000090 75 65 6e 63 68 65 6e 00 00 00 00 00 00 00 00 |uenchen.....
000000a0 24 82 04 08 14 9a 04 08 06 00 00 00 f2 82 04 08 |$.
000000b0 d0 7c 04 40 f4 1f 15 40 01 00 00 00 00 00 00 00 |Þ|.@ð..@.....
000000c0 4d 75 73 74 65 72 6d 61 6e 6e 00 00 00 00 00 00 |Mustermann.....
000000d0 1c ef ff bf a5 90 7e 00 60 ef ff bf 18 75 01 40 |.iÿçÿ.~.`iÿç.u.@
000000e0 c6 82 04 08 20 00 00 00 54 68 6f 6d 61 73 00 00 |.Ä...Thomas..
000000f0 08 db 04 40 70 3a 04 40 a0 7c 01 40 a0 7c 01 40 |.Û.@Þ: @ |. @ |.
00000100 03 00 00 00 58 7f 01 40 88 7f 01 40 f4 81 04 08 |...X...@...@
00000110 4c 6f 74 68 73 74 72 61 73 73 65 00 c4 76 01 40 |Lothstrasse.Äv@
00000120 00 00 00 00 d0 ef ff bf 9c 82 00 40 f4 6f 01 40 |...Ðiÿç...@ð@
00000130 01 00 00 00 18 75 01 40 70 ef ff bf 56 d0 00 40 |...u.@piÿçVð@
00000140 c4 76 38 30 30 30 30 4d 75 65 6e 63 68 65 6e 00 |Äv80000Muenchen.
00000150 00 00 00 00 00 00 00 00 d4 81 04 08 00 9a 04 08 |.....ð.....
00000160 01 00 00 00 c6 82 04 08 70 3a 04 40 f4 1f 15 40 |...Ä...p: @ð..@
00000170 02 00 00 00 00 00 00 00 4d 75 73 74 65 72 6d 61 |.....Musterma
```

Verwaltung von Linux-Dateisystemen (1)

Übersicht

- Dateisystem erzeugen: **mkfs, mkfs.typ**
- Dateisystem auf Konsistenz überprüfen: **fsck, fsck.typ**
- Dateisystem einbinden: **mount**
- Dateisystem entfernen: **umount**
- Spezialoptionen: **reiserfstune, tune2fs, debugfs**
- Informationen: **dumpe2fs**

```
Sep 19 14:27:41 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 01:00:01 amd64 /usr/sbin/cron[29278]: (root) CMD (/sbin/evlogmgr -c 'severity=DEBUG')
Sep 20 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 02:00:01 amd64 /usr/sbin/cron[30103]: (root) CMD (/sbin/evlogmgr -c 'age > *30d')
Sep 20 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 12:46:44 amd64 sshd[6516]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62004
Sep 20 12:46:44 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 12:48:41 amd64 sshd[6609]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62105
Sep 20 12:54:44 amd64 sshd[6694]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62514
Sep 20 15:27:35 amd64 sshd[9077]: Accepted rsa for esser from ::ffff:87.234.201.207 port 64342
Sep 20 15:27:35 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 16:37:11 amd64 sshd[10102]: Accepted rsa for esser from ::ffff:87.234.201.207 port 63375
Sep 20 16:37:11 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 16:38:10 amd64 sshd[10140]: Accepted rsa for esser from ::ffff:87.234.201.207 port 63546
Sep 21 01:00:01 amd64 /usr/sbin/cron[17055]: (root) CMD (/sbin/evlogmgr -c 'severity=DEBUG')
Sep 21 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 21 02:00:01 amd64 /usr/sbin/cron[17878]: (root) CMD (/sbin/evlogmgr -c 'age > *30d')
Sep 21 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 21 17:43:26 amd64 sshd[32081]: Accepted rsa for esser from ::ffff:87.234.201.207 port 63397
Sep 21 17:43:26 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 21 17:53:39 amd64 sshd[3269]: Accepted rsa for esser from ::ffff:87.234.201.207 port 64391
Sep 21 18:43:26 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 21 19:43:26 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 22 01:00:01 amd64 /usr/sbin/cron[4674]: (root) CMD (/sbin/evlogmgr -c 'severity=DEBUG')
Sep 22 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 22 02:00:01 amd64 /usr/sbin/cron[5475]: (root) CMD (/sbin/evlogmgr -c 'age > *30d')
Sep 22 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 22 02:23:21 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 23 01:00:01 amd64 /usr/sbin/cron[24725]: (root) CMD (/sbin/evlogmgr -c 'severity=DEBUG')
Sep 23 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 23 02:00:01 amd64 /usr/sbin/cron[24725]: (root) CMD (/sbin/evlogmgr -c 'severity=DEBUG')
Sep 23 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 23 18:04:05 amd64 sshd[6554]: Accepted publickey for esser from ::ffff:192.168.1.5 port 29771 ssh2
Sep 23 18:04:05 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 23 18:04:26 amd64 sshd[6606]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62093
Sep 24 01:00:01 amd64 /usr/sbin/cron[12436]: (root) CMD (/sbin/evlogmgr -c 'severity=DEBUG')
Sep 24 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 11:15:48 amd64 sshd[20998]: Accepted rsa for esser from ::ffff:87.234.201.207 port 64456
Sep 24 11:15:48 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 13:49:08 amd64 sshd[3197]: Accepted rsa for esser from ::ffff:87.234.201.207 port 61330
Sep 24 13:49:08 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 15:42:07 amd64 kernel: end_seq_mid_events: unsupported module, tainting kernel.
Sep 24 15:42:07 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 15:42:07 amd64 kernel: end_seq_oss: unsupported module, tainting kernel.
Sep 24 20:25:31 amd64 sshd[29399]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62566
Sep 24 01:00:01 amd64 /usr/sbin/cron[662]: (root) CMD (/sbin/evlogmgr -c 'severity=DEBUG')
Sep 25 01:00:02 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 02:00:01 amd64 /usr/sbin/cron[1484]: (root) CMD (/sbin/evlogmgr -c 'age > *30d')
Sep 25 02:00:02 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 10:59:25 amd64 sshd[8889]: Accepted rsa for esser from ::ffff:87.234.201.207 port 64183
Sep 25 10:59:25 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 10:59:47 amd64 sshd[8921]: Accepted rsa for esser from ::ffff:87.234.201.207 port 64253
Sep 25 11:30:02 amd64 sshd[9372]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62029
Sep 25 11:59:25 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 14:05:37 amd64 sshd[11594]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62822
Sep 25 14:05:37 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 14:06:10 amd64 sshd[11586]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62951
Sep 25 14:07:17 amd64 sshd[11608]: Accepted rsa for esser from ::ffff:87.234.201.207 port 63392
Sep 25 14:08:33 amd64 sshd[11630]: Accepted rsa for esser from ::ffff:87.234.201.207 port 63709
Sep 25 15:25:33 amd64 sshd[12930]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62778
```

9.4.1 Verwaltung von Dateisystemen unter Linux

Verwaltung von Linux-Dateisystemen (2)

Dateisystem erzeugen (= formatieren):

mkfs, mkfs.typ

- Beispiel: Ext3-Dateisystem
 - **mkfs.ext3** oder **mkext3fs** (???)
 - Syntax: **mkfs.ext3** [Optionen] [Medium], z. B. **mkfs.ext3 -L Betriebssysteme /dev/hda9**

Verwaltung von Linux-Dateisystemen (3)

```
# mkfs.ext3 -L Betriebssysteme /dev/hda9
mke2fs 1.36 (05-Feb-2005)
Dateisystem-Label=Betriebssysteme
OS-Typ: Linux
Blockgröße=1024 (log=0)
Fragmentgröße=1024 (log=0)
2512 Inodes, 10000 Blöcke
500 Blöcke (5.00%) reserviert für den Superuser
erster Datenblock=1
2 Blockgruppen
8192 Blöcke pro Gruppe, 8192 Fragmente pro Gruppe
1256 Inodes pro Gruppe
Superblock-Sicherungskopien gespeichert in den Blöcken:
      8193

Schreibe Inode-Tabellen: erledigt
Erstelle Journal (1024 Blöcke): erledigt
Schreibe Superblöcke und Dateisystem-Accountinginformationen: erledigt

Das Dateisystem wird automatisch alle 28 Mounts bzw. alle 180 Tage überprüft,
je nachdem, was zuerst eintritt. Veränderbar mit tune2fs -c oder -t .
# _
```

Verwaltung von Linux-Dateisystemen (5)

Dateisystem einbinden: **mount**

- **Syntax:**

```
mount -t Typ -o Optionen Medium Verzeichnis
```

- Verzeichnis muss bereits existieren

- Beispiele:

```
# mount -t ext3 /dev/hda9 /mnt/mnt
```

```
# mount -t vfat /dev/fd0 /mnt/floppy
```

Verwaltung von Linux-Dateisystemen (4)

Dateisystem auf Konsistenz überprüfen: **fsck, fsck.typ**

- nur im nicht gemounteten Zustand
- Beispiel: ReiserFS-Dateisystem
 - **fsck.reiserfs** oder **reiserfsck**
 - **fsck.reiserfs [Optionen] [Medium]**

```
# fsck.ext3 /dev/hda9
e2fsck 1.36 (05-Feb-2005)
Betriebssysteme (/dev/hda9): i.O., 11/2512 Dateien,
1366/10000 Blöcke
```

Verwaltung von Linux-Dateisystemen (6)

- Einträge in **/etc/fstab** definieren Standard-Mounts

# device	mount point	fs	options	dump	fsck
/dev/sda6	/	reiserfs	acl,user_xattr	1	1
/dev/sda1	/windows/C	ntfs	ro,users,gid=users,umask=0002,nls=utf8	0	0
/dev/sda3	/windows/D	vfat	users,gid=users,umask=0002,utf8=true	0	0
/dev/sda4	/windows/E	vfat	users,gid=users,umask=0002,utf8=true	0	0
/dev/sda5	swap	swap	pri=42	0	0
devpts	/dev/pts	devpts	mode=0620,gid=5	0	0
proc	/proc	proc	defaults	0	0
usbfs	/proc/bus/usb	usbfs	noauto	0	0
sysfs	/sys	sysfs	noauto	0	0
/dev/dvd	/media/dvd	iso9660	noauto,nosuid,nodev,exec,iocharset=utf8	0	0
/dev/dvdrecorder	/media/dvdrecorder	iso9660	noauto,nosuid,nodev,exec,iocharset=utf8	0	0

- Mounten solcher Einträge ohne volle Angaben:

```
# mount /media/dvd
# mount /windows/D
```


Verwaltung von Linux-Dateisystemen (7)

Dateisystem entfernen: **umount**

- **umount** entweder den Gerätenamen oder der Mount-Punkt übergeben;
- **umount** findet die Informationen über gemountete Dateisysteme in */etc/mtab*

```
# umount /media/dvd
# umount /windows/D
```

Verwaltung von Linux-Dateisystemen (9)

```
NAME
    tune2fs - adjust tunable filesystem parameters on ext2/ext3 filesystems

SYNOPSIS
    tune2fs [ -l ] [ -c max-mount-counts ] [ -e errors-behavior ] [ -f ] [ -i interval-between-checks ] [ -j ] [ -J journal-options ] [ -m reserved-blocks-percentage ] [ -o [^]mount-options[,...] ] [ -r reserved-blocks-count ] [ -s sparse-super-flag ] [ -u user ] [ -g group ] [ -C mount-count ] [ -L volume-name ] [ -M last-mounted-directory ] [ -O [^]feature[,...] ] [ -T time-last-checked ] [ -U UUID ] device
```

```
DESCRIPTION
    tune2fs allows the system administrator to adjust various tunable filesystem parameters on Linux ext2/ext3 filesystems.
```

```
NAME
    debugfs - ext2/ext3 file system debugger
```

```
SYNOPSIS
    debugfs [ -Vwci ] [ -b blocksize ] [ -s superblock ] [ -f cmd_file ] [ -R request ] [ -d data_source_device ] [ device ]
```

```
DESCRIPTION
    The debugfs program is an interactive file system debugger. It can be used to examine and change the state of an ext2 file system. device is the special file corresponding to the device containing the ext2 file system (e.g /dev/hdXX).
```

Verwaltung von Linux-Dateisystemen (8)

Spezialoptionen: **reiserfstune**, **tune2fs**, **debugfs**

```
NAME
    reiserfstune - The tuning tool for the ReiserFS filesystem.
```

```
SYNOPSIS
    reiserfstune [ -f ] [ -j | --journal-device FILE ] [ --no-journal-available ] [ --journal-new-device FILE ] [ --make-journal-standard ] [ -s | --journal-new-size N ] [ -o | --journal-new-offset N ] [ -t | --max-transaction-size N ] [ -b | --add-badblocks file ] [ -B | --badblocks file ] [ -u | --uuid UUID ] [ -l | --label LABEL ] device
```

```
DESCRIPTION
    reiserfstune is used for tuning the ReiserFS. It can change two journal parameters (the journal size and the maximum transaction size), and it can move the journal's location to a new specified block device. (The old ReiserFS's journal may be kept unused, or discarded at the user's option.) Besides that reiserfstune can store the bad block list to the ReiserFS and set UUID and LABEL. Note: At the time of writing the relocated journal was implemented for a special release of ReiserFS, and was not expected to be put into the mainstream kernel until approximately Linux 2.5. This means that if you have the stock kernel you must apply a special patch. Without this patch the kernel will refuse to mount the newly modified file system. We will charge $25 to explain this to you if you ask us why it doesn't work.
```

Perhaps the most interesting application of this code is to put the journal on a solid state disk.

Verwaltung von Linux-Dateisystemen (10)

Informationen abfragen: **dumpe2fs**

```
# dumpe2fs /dev/hda9
dumpe2fs 1.36 (05-Feb-2005)
Filesystem volume name:   Betriebssysteme
Last mounted on:         <not available>
Filesystem UUID:         2127b3b7-0a24-441f-83c5-30ea11641d85
Filesystem magic number: 0xEF53
Filesystem revision #:   1 (dynamic)
Filesystem features:     has_journal filetype needs_recovery sparse_super
Default mount options:  (none)
Filesystem state:       clean
Errors behavior:        Continue
Filesystem OS type:     Linux
Inode count:            10000
Block count:            500
Reserved block count:   500
Free blocks:            8634
Free inodes:            2501
First block:            1
Block size:             1024
Fragment size:         1024
Blocks per group:      8192
Fragments per group:   8192
Inodes per group:      1256
Inode blocks per group: 157
Filesystem created:    Mon Dec 11 23:59:50 2006
Last mount time:      Tue Dec 12 00:06:37 2006
Last write time:     Tue Dec 12 00:06:37 2006
Mount count:          1
Maximum mount count:   28
Last checked:         Mon Dec 11 23:59:50 2006
Check interval:       15552000 (6 months)
...
```

Verwaltung von Linux-Dateisystemen (11)

- Welche Dateisysteme unterstützt die laufende Linux-Version?
-> Blick in /proc/filesystems
- Linux lädt Dateisystemtreiber (als Module) nach, sobald ein **mount**-Befehl für ein (noch nicht) unterstütztes Dateisystem ausgeführt wird

```
$ cat /proc/filesystems
nodev sysfs
nodev rootfs
nodev bdev
nodev proc
nodev sockfs
nodev debugfs
nodev pipefs
nodev futexfs
nodev tmpfs
nodev eventpollfs
nodev devpts
nodev ext2
nodev ramfs
nodev hugetlbfs
nodev minix
nodev iso9660
nodev nfs
nodev mqueue
nodev rpc_pipefs
nodev reiserfs
nodev usbfs
nodev ntfs
nodev vfat
nodev subfs
nodev hfsplus
```

Vorschau

nächste Vorlesung:

Unter Windows mit Dateien arbeiten