

Prof. Dr. Chr. Vogt

von Kulyk Nazar

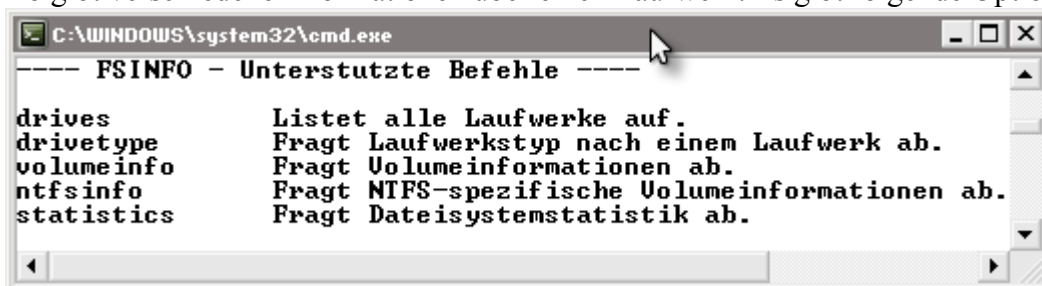
(Matr.Nr.:258360030100)

Ich habe die Arbeit selbständig verfasst, keine anderen als die angegebenen Quellen oder Hilfsmittel benutzt sowie wörtliche und sinnngemäße Zitate als solche gekennzeichnet.

Aufgabe 1.

> Dokumentieren Sie den Unterbefehl `fsutil fsinfo`.

`Fsutil fsinfo` gibt verschiedene Informationen über einen Laufwerk. Es gibt folgende Optionen:



```
C:\WINDOWS\system32\cmd.exe
---- FSINFO - Unterstützte Befehle ----
drives          Listet alle Laufwerke auf.
drivetype       Fragt Laufwerkstyp nach einem Laufwerk ab.
volumeinfo      Fragt Volumeinformationen ab.
ntfsinfo        Fragt NTFS-spezifische Volumeinformationen ab.
statistics      Fragt Dateisystemstatistik ab.
```

Man kann Informationen über alle vorhandenen Laufwerke bekommen (Auflistung)



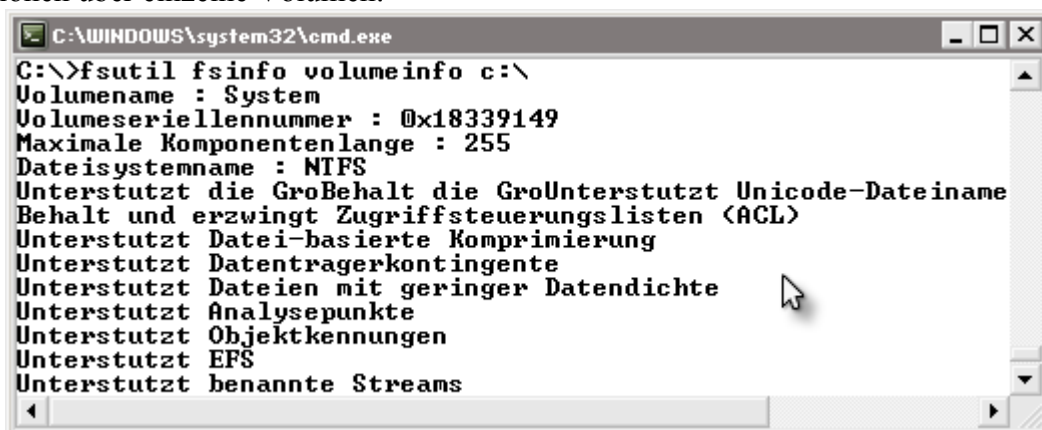
```
C:\WINDOWS\system32\cmd.exe
C:\Dokumente und Einstellungen\gamer>fsutil fsinfo drives
Laufwerke: C:\ D:\ E:\ F:\ G:\
```



```
C:\WINDOWS\system32\cmd.exe
C:\Dokumente und Einstellungen\gamer>fsutil fsinfo drivetype C:
C: - Eingebautes Laufwerk

C:\Dokumente und Einstellungen\gamer>fsutil fsinfo drivetype E:
E: - CD-ROM-Laufwerk
```

Informationen über einzelne Volumes:



```
C:\WINDOWS\system32\cmd.exe
C:\>fsutil fsinfo volumeinfo c:\
Volumename : System
Volumeseriellennummer : 0x18339149
Maximale Komponentenlänge : 255
Dateisystemname : NTFS
Unterstützt die Groß- und Kleinschreibung : Ja
Behält und erzwingt Zugriffsteuerungslisten (ACL) : Ja
Unterstützt Datei-basierte Komprimierung : Ja
Unterstützt Datenträgerkontingente : Ja
Unterstützt Dateien mit geringer Datendichte : Ja
Unterstützt Analysepunkte : Ja
Unterstützt Objektkennungen : Ja
Unterstützt EFS : Ja
Unterstützt benannte Streams : Ja
```

NTFS Spezifische Informationen über einen Volume:

```
C:\WINDOWS\system32\cmd.exe
C:\>fsutil fsinfo ntfsinfo c:
NTFS-Volumeseriennummer : 0x521833af18339149
Version : 3.1
Anzahl der Sektoren : 0x00000000003a962b0
Gesamtzahl Cluster : 0x00000000000752c56
Freie Cluster : 0x000000000001a3961
Insgesamt reserviert : 0x00000000000000000
Bytes pro Sektor : 512
Bytes pro Cluster : 4096
Bytes pro Dateidatensatzsegment : 1024
Cluster pro Dateidatensatzsegment : 0
MFT-gültige Datenlänge : 0x000000000002b98000
MFT-Start-LCN : 0x000000000000c0000
MFT2-Start-LCN : 0x000000000003a962b
MFT-Zonenstart : 0x00000000000490940
MFT-Zoneende : 0x00000000000491ce0
```

Und als letzte Option – Statistik zu einem Volume:

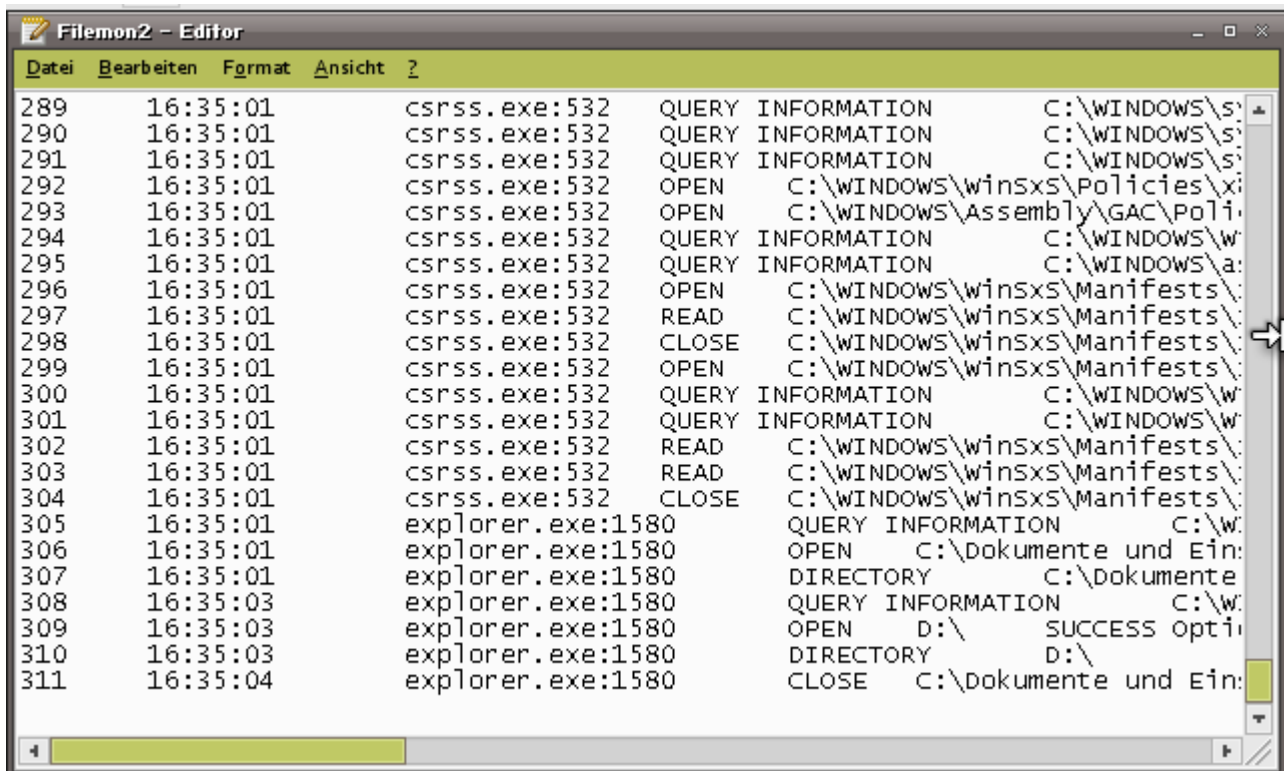
```
C:\WINDOWS\system32\cmd.exe
C:\>fsutil fsinfo statistics c:
Dateisystem : NTFS

UserFileReads : 9235
UserFileReadBytes : 330729472
UserDiskReads : 9218
UserFileWrites : 1932
UserFileWriteBytes : 28409856
UserDiskWrites : 1985
MetaDataReads : 858
MetaDataReadBytes : 30224384
MetaDataDiskReads : 1546
MetaDataWrites : 1127
MetaDataWriteBytes : 7049216
MetaDataDiskWrites : 1612

MftReads : 552
MftReadBytes : 28971008
MftWrites : 995
MftWriteBytes : 6148096
Mft2Writes : 0
Mft2WriteBytes : 0
RootIndexReads : 0
RootIndexReadBytes : 0
RootIndexWrites : 0
RootIndexWriteBytes : 0
BitmapReads : 235
BitmapReadBytes : 962560
BitmapWrites : 110
BitmapWriteBytes : 782336
MftBitmapReads : 2
MftBitmapReadBytes : 8192
MftBitmapWrites : 19
MftBitmapWriteBytes : 106496
UserIndexReads : 687
UserIndexReadBytes : 2813952
UserIndexWrites : 470
UserIndexWriteBytes : 2621440
LogFileReads : 6
LogFileReadBytes : 24576
LogFileWrites : 1227
LogFileWriteBytes : 9207808
```

Aufgabe 2.

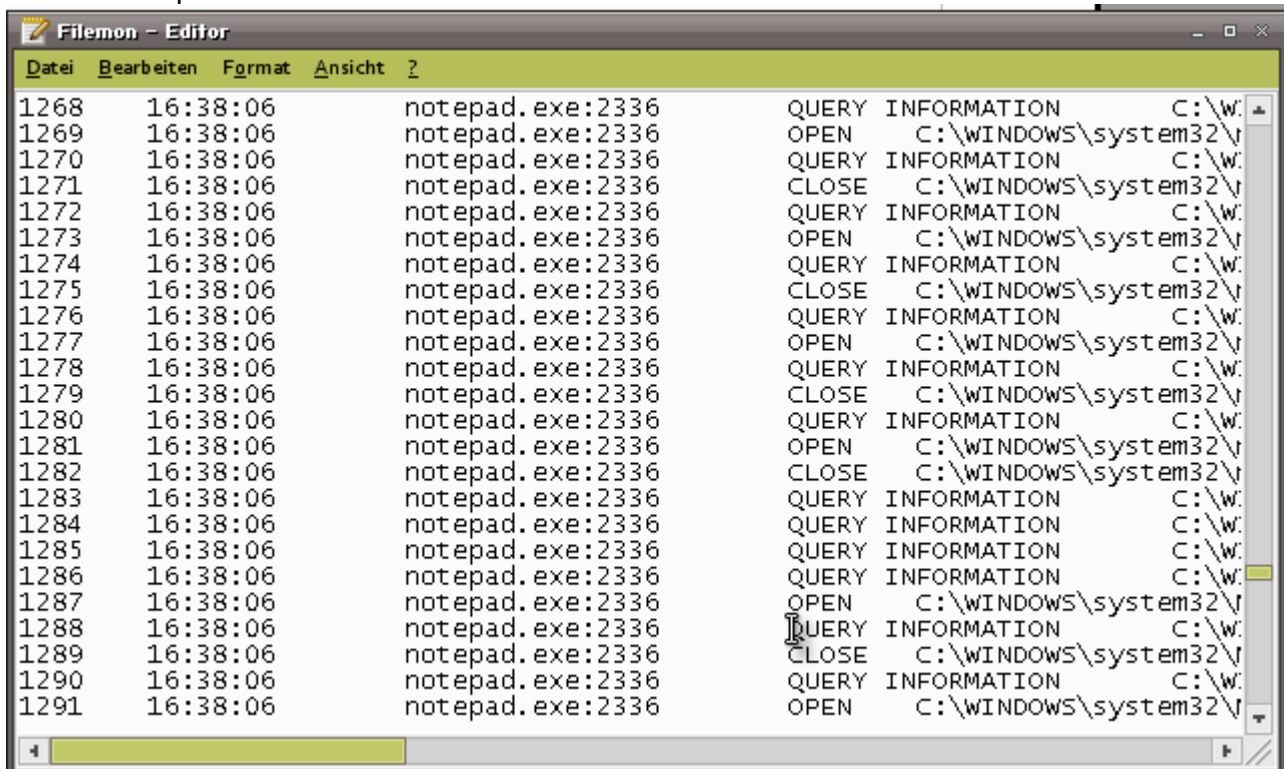
>Starten Sie Filemon und lassen Sie es eine kurze Zeit lang laufen. Speichern Sie die Ausgabe in einer Datei und öffnen Sie diese z.B. mit Notepad. Speichern Sie die Ausgabe erneut, nachdem Sie Notepad gestartet haben, und schauen Sie sich jetzt die neue Datei in Notepad an.



The screenshot shows the Filemon2 - Editor window with a list of system events. The events are categorized by process name, time, and operation type. The processes shown are csrss.exe:532 and explorer.exe:1580. The operations include QUERY INFORMATION, OPEN, DIRECTORY, and CLOSE. The paths involved are primarily C:\WINDOWS\system32 and C:\Dokumente und Einträge.

| Line | Time | Process | Operation | Path |
|------|----------|-------------------|-------------------|---------------------------------------|
| 289 | 16:35:01 | csrss.exe:532 | QUERY INFORMATION | C:\WINDOWS\system32 |
| 290 | 16:35:01 | csrss.exe:532 | QUERY INFORMATION | C:\WINDOWS\system32 |
| 291 | 16:35:01 | csrss.exe:532 | QUERY INFORMATION | C:\WINDOWS\system32 |
| 292 | 16:35:01 | csrss.exe:532 | OPEN | C:\WINDOWS\system32\winxss\Policies\ |
| 293 | 16:35:01 | csrss.exe:532 | OPEN | C:\WINDOWS\system32\Assembly\GAC\Pol |
| 294 | 16:35:01 | csrss.exe:532 | QUERY INFORMATION | C:\WINDOWS\system32 |
| 295 | 16:35:01 | csrss.exe:532 | QUERY INFORMATION | C:\WINDOWS\system32 |
| 296 | 16:35:01 | csrss.exe:532 | OPEN | C:\WINDOWS\system32\winxss\Manifests\ |
| 297 | 16:35:01 | csrss.exe:532 | READ | C:\WINDOWS\system32\winxss\Manifests\ |
| 298 | 16:35:01 | csrss.exe:532 | CLOSE | C:\WINDOWS\system32\winxss\Manifests\ |
| 299 | 16:35:01 | csrss.exe:532 | OPEN | C:\WINDOWS\system32\winxss\Manifests\ |
| 300 | 16:35:01 | csrss.exe:532 | QUERY INFORMATION | C:\WINDOWS\system32 |
| 301 | 16:35:01 | csrss.exe:532 | QUERY INFORMATION | C:\WINDOWS\system32 |
| 302 | 16:35:01 | csrss.exe:532 | READ | C:\WINDOWS\system32\winxss\Manifests\ |
| 303 | 16:35:01 | csrss.exe:532 | READ | C:\WINDOWS\system32\winxss\Manifests\ |
| 304 | 16:35:01 | csrss.exe:532 | CLOSE | C:\WINDOWS\system32\winxss\Manifests\ |
| 305 | 16:35:01 | explorer.exe:1580 | QUERY INFORMATION | C:\WINDOWS\system32 |
| 306 | 16:35:01 | explorer.exe:1580 | OPEN | C:\Dokumente und Einträge |
| 307 | 16:35:01 | explorer.exe:1580 | DIRECTORY | C:\Dokumente und Einträge |
| 308 | 16:35:03 | explorer.exe:1580 | QUERY INFORMATION | C:\WINDOWS\system32 |
| 309 | 16:35:03 | explorer.exe:1580 | OPEN | D:\SUCCESS Opti |
| 310 | 16:35:03 | explorer.exe:1580 | DIRECTORY | D:\ |
| 311 | 16:35:04 | explorer.exe:1580 | CLOSE | C:\Dokumente und Einträge |

Und mit Notepad laufend:



The screenshot shows the Filemon - Editor window with a list of system events for notepad.exe:2336. The events are categorized by time and operation type. The operations include QUERY INFORMATION, OPEN, and CLOSE. The paths involved are primarily C:\WINDOWS\system32.

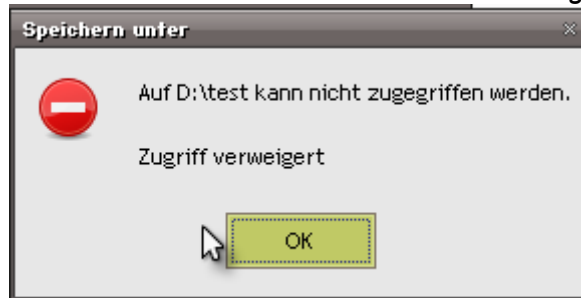
| Line | Time | Process | Operation | Path |
|------|----------|------------------|-------------------|---------------------|
| 1268 | 16:38:06 | notepad.exe:2336 | QUERY INFORMATION | C:\WINDOWS\system32 |
| 1269 | 16:38:06 | notepad.exe:2336 | OPEN | C:\WINDOWS\system32 |
| 1270 | 16:38:06 | notepad.exe:2336 | QUERY INFORMATION | C:\WINDOWS\system32 |
| 1271 | 16:38:06 | notepad.exe:2336 | CLOSE | C:\WINDOWS\system32 |
| 1272 | 16:38:06 | notepad.exe:2336 | QUERY INFORMATION | C:\WINDOWS\system32 |
| 1273 | 16:38:06 | notepad.exe:2336 | OPEN | C:\WINDOWS\system32 |
| 1274 | 16:38:06 | notepad.exe:2336 | QUERY INFORMATION | C:\WINDOWS\system32 |
| 1275 | 16:38:06 | notepad.exe:2336 | CLOSE | C:\WINDOWS\system32 |
| 1276 | 16:38:06 | notepad.exe:2336 | QUERY INFORMATION | C:\WINDOWS\system32 |
| 1277 | 16:38:06 | notepad.exe:2336 | OPEN | C:\WINDOWS\system32 |
| 1278 | 16:38:06 | notepad.exe:2336 | QUERY INFORMATION | C:\WINDOWS\system32 |
| 1279 | 16:38:06 | notepad.exe:2336 | CLOSE | C:\WINDOWS\system32 |
| 1280 | 16:38:06 | notepad.exe:2336 | QUERY INFORMATION | C:\WINDOWS\system32 |
| 1281 | 16:38:06 | notepad.exe:2336 | OPEN | C:\WINDOWS\system32 |
| 1282 | 16:38:06 | notepad.exe:2336 | CLOSE | C:\WINDOWS\system32 |
| 1283 | 16:38:06 | notepad.exe:2336 | QUERY INFORMATION | C:\WINDOWS\system32 |
| 1284 | 16:38:06 | notepad.exe:2336 | QUERY INFORMATION | C:\WINDOWS\system32 |
| 1285 | 16:38:06 | notepad.exe:2336 | QUERY INFORMATION | C:\WINDOWS\system32 |
| 1286 | 16:38:06 | notepad.exe:2336 | QUERY INFORMATION | C:\WINDOWS\system32 |
| 1287 | 16:38:06 | notepad.exe:2336 | OPEN | C:\WINDOWS\system32 |
| 1288 | 16:38:06 | notepad.exe:2336 | QUERY INFORMATION | C:\WINDOWS\system32 |
| 1289 | 16:38:06 | notepad.exe:2336 | CLOSE | C:\WINDOWS\system32 |
| 1290 | 16:38:06 | notepad.exe:2336 | QUERY INFORMATION | C:\WINDOWS\system32 |
| 1291 | 16:38:06 | notepad.exe:2336 | OPEN | C:\WINDOWS\system32 |

> Schauen Sie sich auch an, welche Dateiaktivitäten durch den Start von Notepad ausgelöst wurden.

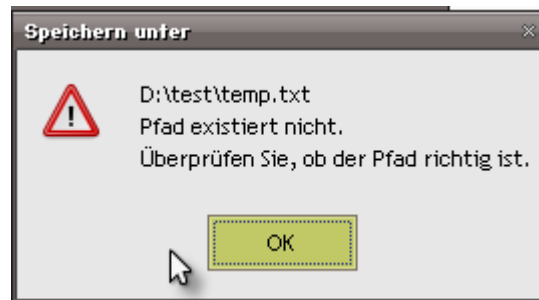
Falls Notepad gestartet wird, werden zusätzlich noch DLLs geladen werden. Dabei gibt es erst eine QUERY Anfrage, die Datei Attribute sich holt und dann mit OPEN wird die Datei geladen. Dannach kommt einen CLOSE Event.
SET INFORMATION Speichert in einen User bezogenen Log Informationen.

Aufgabe 3.

>- „Browsen“ Sie zunächst zu dem Verzeichnis. Welche Fehlermeldung erscheint?



> Geben Sie nun im „Save As“-Dialog den kompletten Pfad selbst ein. Die Fehlermeldung kann ja wohl nicht stimmen ... !



| # | Time | Process | Request | Path | Result | Other |
|----|----------|--------------|-----------|-------------------|-------------|----------------------------------|
| 1 | 17:02:34 | notepad.e... | OPEN | D:\ | SUCCESS | Options: Open Directory Ac... |
| 2 | 17:02:34 | notepad.e... | DIRECTORY | D:\ | SUCCESS | FileBothDirectoryInformation: * |
| 3 | 17:02:34 | notepad.e... | DIRECTORY | D:\ | SUCCESS | FileBothDirectoryInformation |
| 4 | 17:02:34 | notepad.e... | DIRECTORY | D:\ | NO MORE ... | FileBothDirectoryInformation |
| 5 | 17:02:34 | notepad.e... | CLOSE | D:\ | SUCCESS | |
| 6 | 17:02:36 | notepad.e... | OPEN | D:\ | SUCCESS | Options: Open Directory Ac... |
| 7 | 17:02:36 | notepad.e... | DIRECTORY | D:\ | SUCCESS | FileBothDirectoryInformation:... |
| 8 | 17:02:36 | notepad.e... | CLOSE | D:\ | SUCCESS | |
| 9 | 17:02:36 | notepad.e... | OPEN | D:\test\ | ACCESS D... | XTOWER\gamer |
| 10 | 17:02:42 | notepad.e... | OPEN | D:\test\test.txt | NOT FOUND | Options: Open Directory Ac... |
| 11 | 17:02:42 | notepad.e... | OPEN | D:\test | ACCESS D... | XTOWER\gamer |
| 12 | 17:02:42 | notepad.e... | OPEN | D:\ | SUCCESS | Options: Open Directory Ac... |
| 13 | 17:02:42 | notepad.e... | DIRECTORY | D:\ | SUCCESS | FileBothDirectoryInformation:... |
| 14 | 17:02:42 | notepad.e... | CLOSE | D:\ | SUCCESS | |
| 15 | 17:02:42 | notepad.e... | OPEN | D:\test\ | ACCESS D... | XTOWER\gamer |
| 16 | 17:02:42 | notepad.e... | OPEN | D:\test\test.txt\ | NOT FOUND | Options: Open Directory Ac... |

9 17:02:36 notepad.exe:256 OPEN D:\test\ ACCESS DENIED
 10 17:02:42 notepad.exe:256 OPEN D:\test\test.txt NOT FOUND Options: Open Directory Access: 00100020

Result = NOT FOUND – dass wegen kommt die Fehler Meldung das ein Pfad nicht existiert.

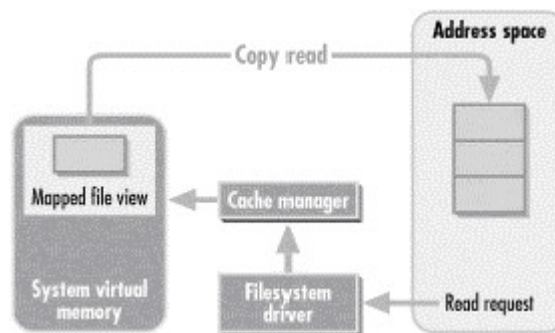
In Other steht: **Options: Open Directory Access: 00100020** – in Wirkligkeit Grund für die Fehler ist einen Access auf Directory.

Aufgabe 4.

[1] Dateisystem von Windows NT (NTFS) ist ein transaktionsorientiertes Dateisystem (journaling), welches alle Verwaltungsdaten als normale Dateien behandelt und speichert. Das wird auch beim Cache deutlich: im Vergleich zu anderen Dateisystemen besitzt Windows NT einen eigenständigen Cache-Manager, der seine Daten auf Basis virtueller Blöcke verwaltet. Diese beinhalten Dateiblöcke, d.h. im Unterschied zum Blockcache werden alle Daten dateiorientiert zwischengespeichert. Durch die Behandlung und Speicherung der Meta-Informationen als normale Dateien ist hier keine getrennte Betrachtung der unterschiedlichen Typen notwendig. Zu jeder Datei – auch Verwaltungsdateien – existieren ein oder mehrere 64bit große Filerecords, diese enthalten die Informationen über die Datei. Der NT-Cache Manager unterstützt das Verankern (pinning, locken) von Speicher. Im Normalfall kann der vom Dateicache verwendete Speicher ausgelagert werden. Der Cache-Manager unterstützt vier verschiedene Zugriffsarten auf Daten:

- Strommanipulation (File stream manipulation).
- Kopieren vom Puffer in Nutzeradressraum (Copy Interface): diese Schnittstelle stellt die Möglichkeiten von Vorauslesen (read ahead) und verzögertem Schreiben (write throttling) zur Verfügung.
- Speicherlisten (Memory Descriptor List): dadurch kann direkt auf den Speicherbereich des Caches zugegriffen werden (direct memory access (DMA)). Die Schnittstelle benutzt die gleichen Funktionsaufrufe für das Vorauslesen wie das Copy Interface.
- Pinning Schnittstelle: stellt einen nichtauslagerbaren Speicherbereich zur Verfügung.

[2] Im Normalfall wenn eine File Handle geöffnet wird im User Mode, wird ein File Object im Kernel Mode erzeugt, wo die Information von NTFS File drinnen steht. Dort wird auch gesagt ob die Datei Buffered oder NonBuffered Zugriff haben soll. In der Regel wird einen Buffered Zugriff gerichtet, dafür gibt es im Kernel Chaching Manager was stark mit Kernel Speicher Verwaltung zusammenhängt.



File Cache ist ein reserviertes Virtuelles Adressraum (VA) im Kernel der bei 0x8000000 anfängt, der auf Chunks 256K aufgeteilt ist. Und jedes Dateiteil, das gekescht wird auf ein Offset für ein Chunk gemapt. Wo dann die Daten sind.

Es gibt Private- und Shared Cache Memory für jede Datei. Shared Memory wird oft bei einem gemeinsamem Zugriff auf eine Datei von verschiedenen Processen benutzt. Und Private Cache Memory wird benutzt bei Read Ahead.

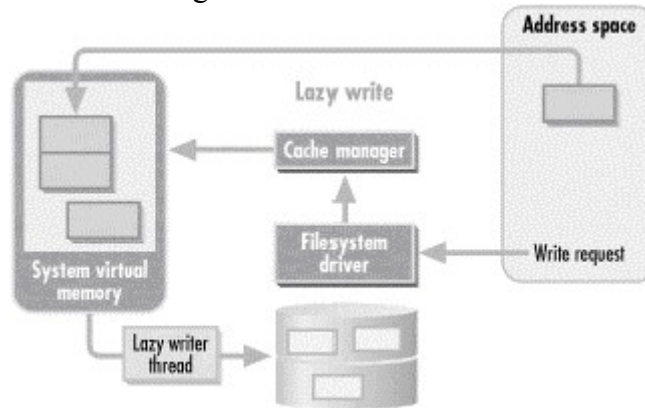
Speicherverwaltung und Cache Manager benutzen in Prinzip die selben Infrastrukturen. Beide können gegenseitig bestimmte Funktionen ausführen. Wenn Cache Manager Speicher für eine neue Datei im Cache VA braucht, oder wenn Speicherverwaltung braucht mehr freier Speicher für Laufende Prozesse.

1 Internet Quelle von 06. März 2000 - http://os.inf.tu-dresden.de/papers_ps/lutter-diplom.pdf

2 Interview with Molly Brown from Microsoft Kernel Team

CacheManager versucht auch das Verhalten von der Application zu erkennen um einen Read Ahead zu machen (das kann auch von benutzer gesteuert werden). Windows FS kann senqenzeil Read Ahead machen, oder rückwärts oder alle X bytes.

[1]Cache Manager startet periodisch einen Flush Vorhang (Lazy Writing). Die veränderten Chunks aus VA werden ein nach dem anderem geschrieben auf die Platte.



In einem 64Bit System gewinnt ein File Cache Manager gerade viel an Performance. Wieso ist das so.

Bei einem 32Bit System wird das VA auf 2Gb beschenkt. Und wenn ein System viel verschiedene Zugriffe auf verschiedene Dateien macht – muss es oft Cache Pages Wegschieben um Speicher frei zu bekommen und wieder zurück holen, wenn die vorige Datei wieder zugegriffen wird. Das nimmt viel Zeit und beeinflusst die Performance von solchem System. Bei 64Bit Systemen ist VA viel größer und da muss dieses Tauschen von VA Chunks nicht so oft passieren.

Allerdings 64Bit Cache Manager braucht einen kompletten Redesign zu vergleicht zu einem 32Bit.

Als Programmierer hat man verschiedene Möglichkeiten aus User Mode (WIN32 API) einen Einfluss auf das Caching-Verhalten zu bekommen. Wir haben schon verschiedene Wege beim bekommen von File Handle auf eine Datei in vorigen Aufgaben beschrieben (CreateFile). Aber es gibt auch noch andere Wege. Ein solches Weg sind „Temporary Files“.

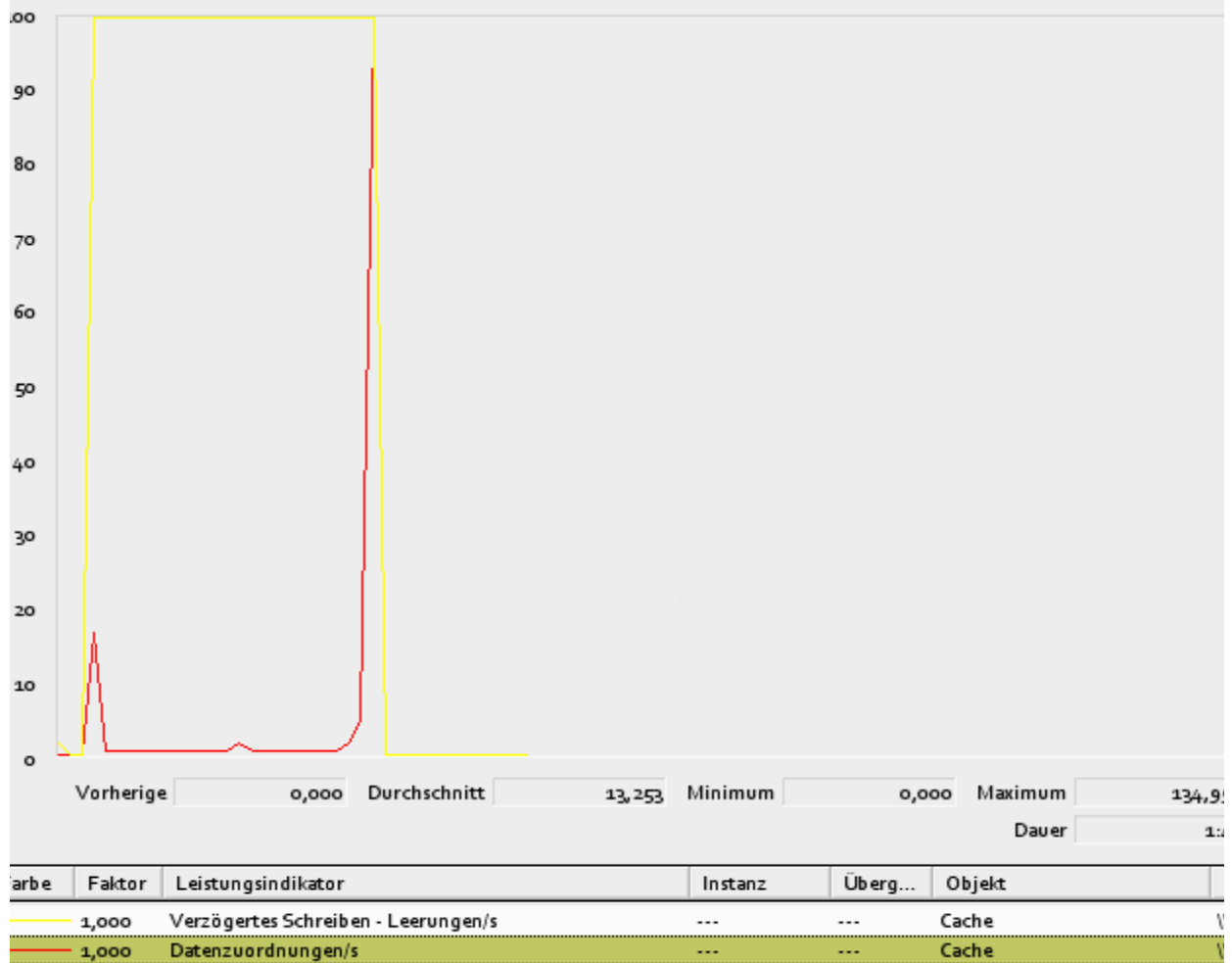
File Handles auf Temporär Files werden mit einem extra Befehl gemacht – GetTempFileName nach einem CreateFile. Diese Dateien werden nie von dem CacheManager auf die Festplatte geschrieben (Flush). Also bei dem Lazy Writting Flush werden alle veränderten Chunks die zu solcher Datei gehören übersprungen.

Verhalten von File Cache Manager kann man auch durch „Sparse Files“ beeinflussen.

1 Internet Quelle von Januar 2002 -

<http://www.microsoft.com/technet/prodtechnol/windows2000serv/maintain/optimize/wperfch7.msp#ETH>

Aufgabe 5.



Am Anfang von dem Kopieren Vorhang werden sofort Alle Daten aus Datei auf der Zielplatte zugeordnet. Da sieht man das es auf 100 geht (Gelbe Linie).
 Das schreiben auf die Festplatte (Flushen) fängt auch langsam an, aber dann erst nicht so oft ausgeführt. Ein mal zwischen werden Daten wider geflusht und erst kurz davor die Ganze Datei Datenzuordnung bekommt werden wirklich alle Daten aus Datei direkt auf die Platte geschrieben. Das Kopieren Vorhang ist zu ende wenn das die Gelbe Linie runter geht. Daten werden auf die Platte geschrieben.