

Mag. iur. Dr. techn. Michael Sonntag

KV Betriebssysteme

**Taskmanager / top
Performance Monitoring**

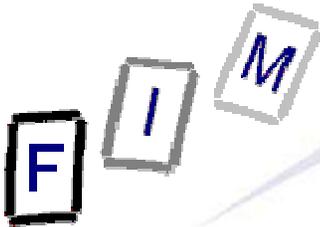
Institut für Informationsverarbeitung und
Mikroprozessortechnik (FIM)
Johannes Kepler Universität Linz, Österreich

E-Mail: sonntag@fim.uni-linz.ac.at
<http://www.fim.uni-linz.ac.at/staff/sonntag.htm>

F I M

Fragen?

Bitte gleich stellen!



Taskmanager

- Einzelne Prozesse können beendet werden
- Prozessstruktur beenden:
 - Auch alle weiteren direkt oder indirekt von diesem Prozess erzeugten Prozesse werden beendet!
 - Beispiel: Outlook → MAPI Spooler wird auch beendet
- Priorität einstellen
 - **Nur CPU**, nicht Speicher, etc.!
- Anzuzeigende Spalten können konfiguriert werden
- Updategeschwindigkeit in 4 Stufen wählbar (Einfluß auf Ergebnis?)

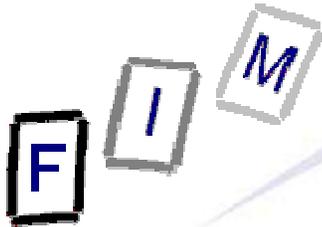
Speicher: Shared pages werden mehrfach gezählt!

The screenshot shows the Windows Task Manager Performance tab. The 'Performance' tab is selected, and the 'Memory' section is expanded. The 'Mem' column in the process list is highlighted. A context menu is open over the 'AcroRd32.exe' process, showing options like 'End Process', 'End Process Tree', 'Debug', and 'Set Priority'. The 'Set Priority' option is selected, and a sub-menu is open showing priority levels: Realtime, High, AboveNormal, Normal (selected), BelowNormal, and Low. A red arrow points from the text above to the 'Mem' column header. Another red arrow points from the text above to the 'Show processes from all users' checkbox at the bottom of the window. A third red arrow points from the text above to the 'End Process' button at the bottom right of the window.

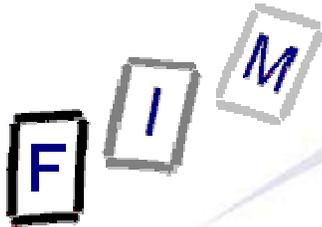
Image Name	User Name	CPU	CPU Time	Mem
EditPad.exe	SONNTAG	00	0:00:02	6
AcroRd32.exe	SONNTAG	00	0:00:00	28
svchost.exe	SYSTEM	00	0:00:00	3
CodedDrag.o	SONNTAG	00	0:00:00	2
POWERPNT.	SONNTAG	00	0:00:41	9
WINWORD.E	SONNTAG	00	0:00:03	20
SYMPROXY.S	SONNTAG	00	0:00:00	7
NISSERV.EXE	SYSTEM	00	0:00:00	4
Eudora.exe	SONNTAG	00	0:00:00	16
SETI@home.exe	SONNTAG	00	0:00:00	17
ctfmon.exe	SONNTAG	00	0:00:00	2
pdesk.exe	SONNTAG	00	0:00:00	3
SMTray.exe	SONNTAG	00	0:00:00	2
imonNT.exe	SYSTEM	00	0:00:00	1
spamnix.exe	SONNTAG	00	0:00:00	13
NISUM.EXE	SYSTEM	00	0:00:00	3
mgabg.exe	SYSTEM	00	0:00:00	1
mdm.exe	SYSTEM	00	0:00:00	3

Für Mehrbenutzer-Rechner

Virtueller Speicher, Performance Monitoring

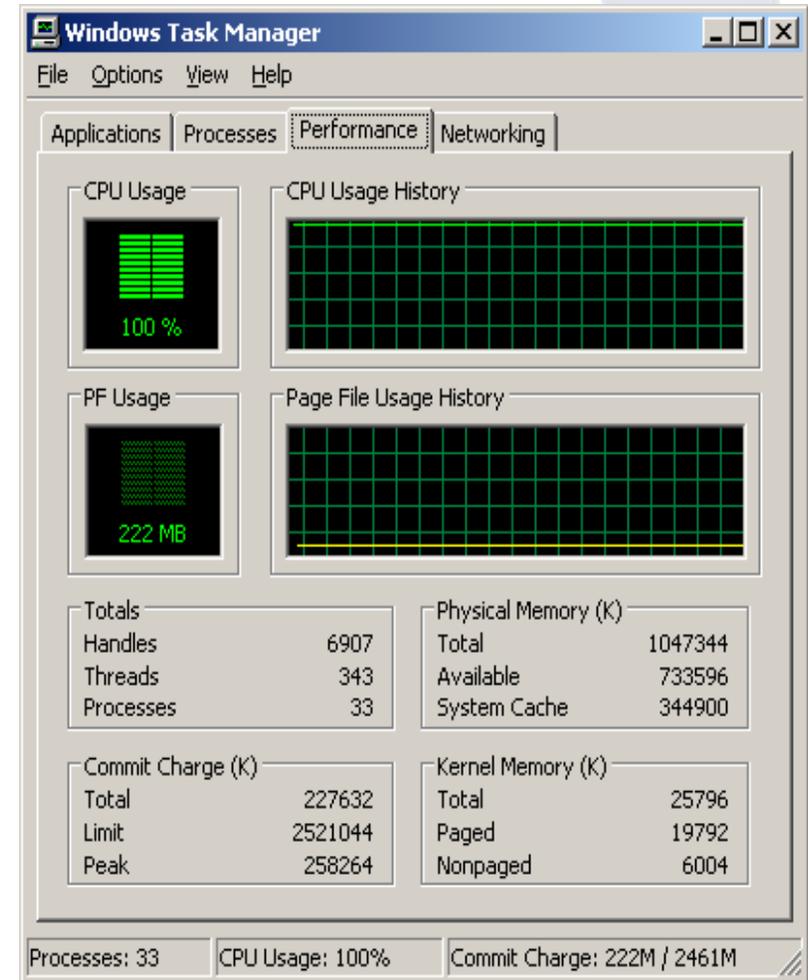


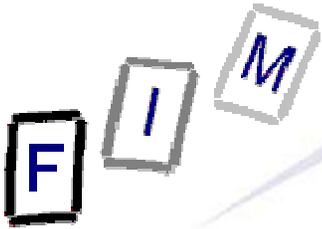
- Spalten können frei ausgewählt werden (Menü)
 - CPU Usage: Anteil an CPU-Nutzung seit der letzten Neuanzeige
 - CPU Time: Absolute verbrauchte CPU Zeit seit dem Start
 - Memory Usage: Working Set in Kilobytes (=RAM)
 - Memory Usage Delta: Unterschied zum vorigen Wert
 - Peak Memory Usage: Höchstwert von Memory Usage
 - Virtual Memory Size: Zugeteilter Adressraum (=Festplatte;exkl. RAM)
 - Page Faults: Anzahl der nötigen Seiten-Einlagerungen seit dem Start
 - Page Faults Delta: Unterschied zum vorigen Wert
 - Paged Pool: Menge des "Arbeits-"Speichers des Prozesses. Dieser kann ausgelagert werden
 - Non-Paged Pool: Menge des Systemspeichers des Prozesses. Dieser kann nicht ausgelagert werden!
 - » PCB, Verwaltungsinformationen, ...
 - ...



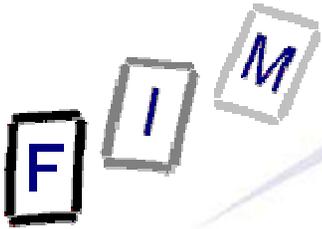
Leistungsstatistiken

- CPU Usage: Derzeitige Auslastung
- PF Usage: Auslastung der Auslagerungsdatei (belegte Menge)
 - Beides mit Statistik über die Zeit
- Commit Charge:
 - Von Prozessen belegter Speicher
 - Limit: Fehler, falls mehr benötigt wird!
 - » =RAM+maximale Pagefile-Größe
- Physical Memory: RAM
 - System Cache: Für offene Dateien, etc.
- Kernel Memory: Vom OS und Treibern belegt
 - Paged: Kann ausgelagert werden
 - Nonpaged: Niemals ausgelagert





- Startet man ein großes Programm (Word, Entwicklungsumgebung, ...) und beobachtet die Statistiken: Wo ändert sich etwas?
 - » CPU-Auslastung steigt und Auslagerungsdatei steigt
 - Und was passiert nachher wenn Sie nichts mit dem Programm tun?
 - » CPU-Auslastung sinkt wieder, Auslagerungsdatei bleibt gleich
- Welche Teile des Betriebssystems fallen unter die Statistik "Nonpaged Kernel Memory"?
 - Die Teile zum Laden von Seiten von der Festplatte
 - Scheduler (und restlicher Betriebssystemkern; Code)
 - Verwaltungsdaten (PCB, Seitentabellen!?!)
 - Gerätetreiber (Code+Daten; nicht alle!)



Einfache Linux Statistiken: "top"

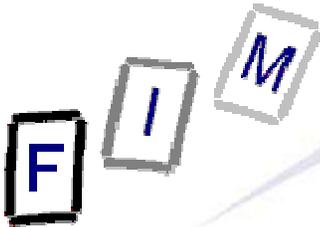
- Einfaches Textmodus-Tool zur Systembeobachtung
 - Funktion ähnlich dem Taskmanager
- Liefert Überblickswerte zu
 - CPU: User, System, IO-Wartezustand und Idle –Anteil
 - Speicher: Benutzt, frei, shared, Buffer
 - Virtueller Speicher: Benutzt, frei
 - Laufende Prozesse und zugehörige Daten
- Veränderungen ebenso möglich:
 - Prozesse beenden
 - Priorität ändern
 - Sortierung der Anzeige
 - Konfiguration der anzuzeigenden Spalten



"top"-Beispiel

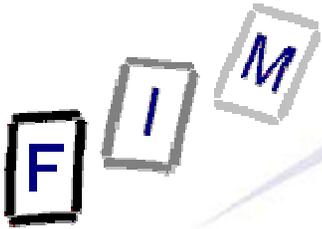
```
12:27:23 up 16 days, 16:54, 1 user, load average: 0,04, 0,06, 0,01
63 processes: 62 sleeping, 1 running, 0 zombie, 0 stopped
CPU states: 1,1% user 2,9% system 0,0% nice 0,0% iowait 95,8% idle
Mem: 248768k av, 201012k used, 47756k free, 0k shrd, 87112k buff
      97512k active, 63684k inactive
Swap: 522104k av, 10848k used, 511256k free 43112k cached
```

PID	USER	PRI	NI	SIZE	RSS	SHARE	STAT	%CPU	%MEM	TIME	CPU	COMMAND
9	root	17	0	0	0	0	SW	1,1	0,0	25:35	0	kjournald
25053	root	18	0	1204	1204	828	R	0,3	0,4	0:00	0	top
1	root	4	0	468	440	420	S	0,0	0,1	0:08	0	init
2	root	9	0	0	0	0	SW	0,0	0,0	0:00	0	keventd
3	root	19	19	0	0	0	SWN	0,0	0,0	0:25	0	ksoftirqd_CPU
4	root	9	0	0	0	0	SW	0,0	0,0	0:33	0	kswapd
5	root	9	0	0	0	0	SW	0,0	0,0	0:00	0	bdflood
6	root	9	0	0	0	0	SW	0,0	0,0	0:03	0	kupdated
62	root	9	0	0	0	0	SW	0,0	0,0	0:00	0	khubd
1592	root	9	0	0	0	0	SW	0,0	0,0	0:00	0	kjournald
1990	root	9	0	0	0	0	SW	0,0	0,0	0:00	0	eth1
3937	named	9	0	3984	3764	1696	S	0,0	1,5	2:09	0	named
3951	root	9	0	852	716	632	S	0,0	0,2	0:01	0	sshd
3965	root	8	0	796	716	648	S	0,0	0,2	0:05	0	xinetd
3975	root	9	0	372	316	300	S	0,0	0,1	0:02	0	vsftpd
3986	root	9	0	1428	972	764	S	0,0	0,3	0:00	0	dhcpcd



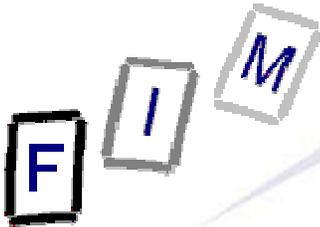
"top"-Spalten

- PID: Prozess-ID
- USER: Username des Besitzers
- PRI: Priorität
- NI: Nice-Wert (negativ = höhere Priorität)
- SIZE: Code+Stack+Heap in KB
- RSS: Belegter RAM in KB
- SHARE: Gemeinsamer Speicher in KB
- STAT: Zustand (S=Sleeping, R=Running, T=Traced, ...)
- %CPU: CPU-Anteil seit dem letzten Refresh
- %MEM: Anteil am RAM
- TIME: Gesamte CPU Zeit seit dem Start des Prozesses
- CPU: Auf welcher CPU ausgeführt
- COMMAND: Name des Kommandos (=Programmname)



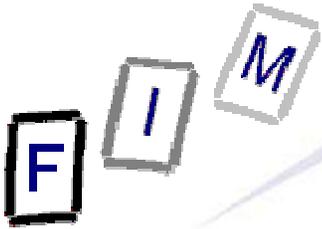
Wozu Performance Monitoring?

- Zukunftsvorhersage auf Basis der Vergangenheit
 - Werden wir neue Festplatten benötigen?
 - Bringt mehr Speicher etwas oder hilft das nichts?
 - Höhere Bandbreite sinnvoll?
- Überprüfen der Veränderungen
 - Hat die neue Festplatte etwas gebracht?
 - Ist es jetzt vielleicht schlechter als vorher?
- Systemkapazität feststellen
 - Was können wir noch anbieten ohne das System zu überlasten?
- Abrechnung
 - Z. B. bei Bezahlung nach Belastung



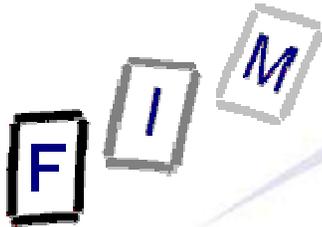
Wozu Performance Monitoring?

- Warnungen bei tatsächlichen Problemen
 - Auslagerungsdatei ist voll
- Feststellen wo die tatsächlichen Engpässe liegen
 - Ist es die Festplatte oder die Festplattenschnittstelle?
- Harte Fakten über den tatsächlichen Zustand
 - » Benutzer klagen oft über Schwierigkeiten, diese sollte man nachprüfen
 - Nicht: "Es geht so langsam"
 - Sondern: "Eine Anfrage benötigt 254 ms"
- Dokumentation der Vergangenheit
- Angriffe erkennen
 - Ungewöhnliches Systemverhalten kann Zeichen eines Angriffes sein
 - » Plötzliche Spitzen der Netzwerkbelastung
 - » Ungewöhnliche neue Prozesse



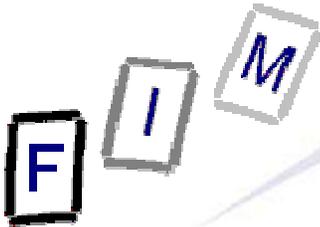
Was sollte man beobachten?

- CPU / CPUs
- Festplatte
 - Einzelne Disks: Fehler auf der Disk kündigen sich an
 - Mehrere Disks: Tatsächliche Leistung
- Speicher:
 - RAM: Zu wenig vorhanden (→viele Pagefaults, Seitenflattern!)
 - Virtual memory: Ausreichend vorhanden, schnell genug, ...
- Netzwerk:
 - Insbes. Kollisionen, Fehler: Hinweis auf schlechte Kabel oder (bald) defekte Hardware
- Bestimmte Anwendungen oder Dienste
 - Laufen sie noch? Besondere Daten (z. B. Zugriffsstatistik bei Webservern, ...)
- Buffer und Caches: Bringen sie etwas, Größe, ...



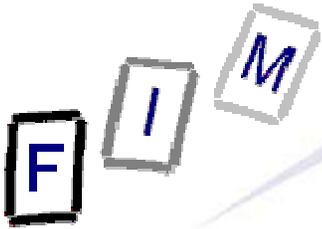
Gibt es dabei Probleme?

- Jede Beobachtung beeinflusst das zu beobachtende System in seinem Verhalten (Siehe Heisenberg!)
 - Beobachtung ist nicht "gratis": Auch sie braucht Rechenzeit und Ressourcen (Speicher, Bildschirm, Festplatte, ...)!
- Meistens relativ niedrig, außer:
 - Graphische Darstellung
 - » Besser reine Text-Logs und später oder auf einem anderem Computer auswerten
 - Kurze Intervalle
 - » Nicht zu oft Werte feststellen: Muß die Speicherauslastung wirklich jede Sekunde festgestellt werden?
 - » Auch: Datenmenge!
 - Sehr viele Dinge werden überwacht
 - » Man kann fast alles überwachen
 - » ABER: System-Beeinträchtigung und **wer wertet das aus!?!**



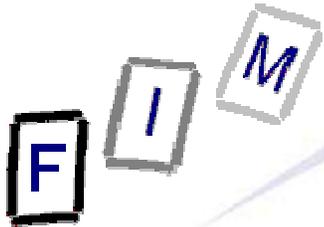
Wie erfolgt die Beobachtung?

- Als eigener Dienst (daemon) der Daten sammelt
 - System-Log oder besonderer Service
- Oder jedes Programm macht dies selbst
 - Eigenes ("privates") Log
 - » Wichtig für Softwareentwicklung (Debugging)
 - » Oft Schnittstellen für "Umleitung" in das System-Log
 - Wichtig für Auslieferungs-Version
- Quelle muß Daten selbst zur Verfügung stellen:
 - Kernel: Bietet sehr viele Daten an
 - Anwendungen: Was sie eben preisgeben
 - » Ansonsten ist man auf die System-Daten angewiesen, aus denen man ev. auf das Programmverhalten schließen kann!
- Technisch: Aufruf eines besonderen Statements, welches die Nachricht (einen String) an den Logger übergibt
 - Spezielle Methoden zur Geschwindigkeitssteigerung
 - » Z.B.: `if (log.isDebugEnabled()) log.debug("..."+"..."+"..."+"...");`

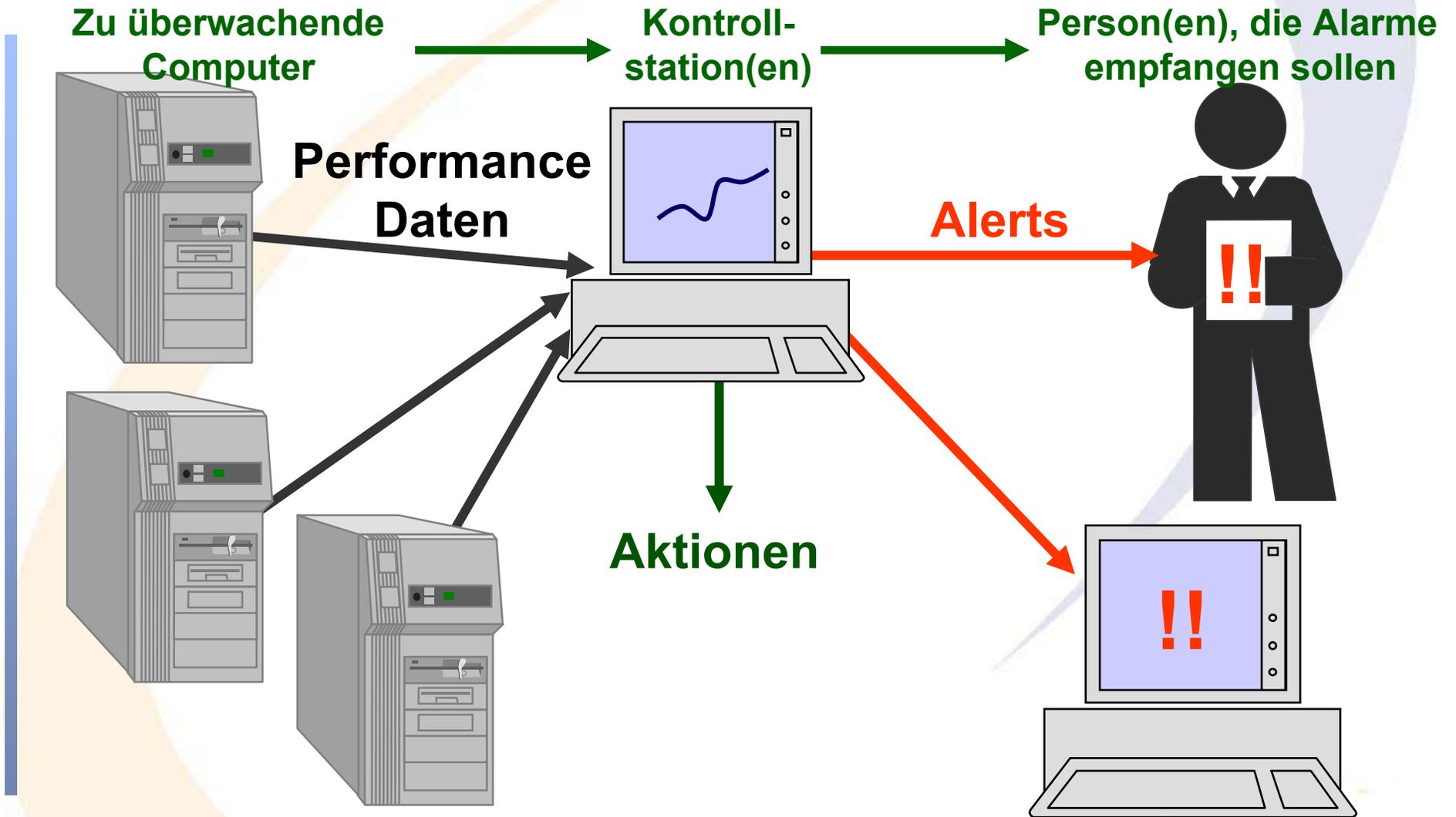


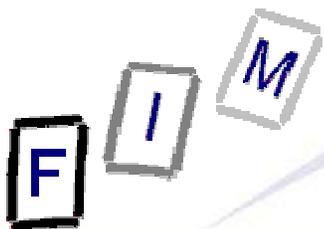
Ausgabearten

- Diagramme (graphischer Überblick)
 - Gut für Zahlen, welche über die Zeit aufgetragen werden
- Performance Logs (zum Archivieren)
 - Nachweis der Funktion/Antwortzeit, etc.
- Reports (einfacher Überblick)
 - Kurz-Zusammenfassung für die regelmäßige Kontrolle
- Alerts: Alarm bei aktuellen Problemen
 - Konkretes, aktuelles und einzelnes Problem
- Dies kann alles automatisch gestartet werden
 - Nach vorhergehender Konfiguration
 - Erste Ansätze z.B. zur automatischen Erkennung von Alarmen
- Dies muß nicht lokal am betroffenen Computer sondern kann auch über das Netzwerk erfolgen

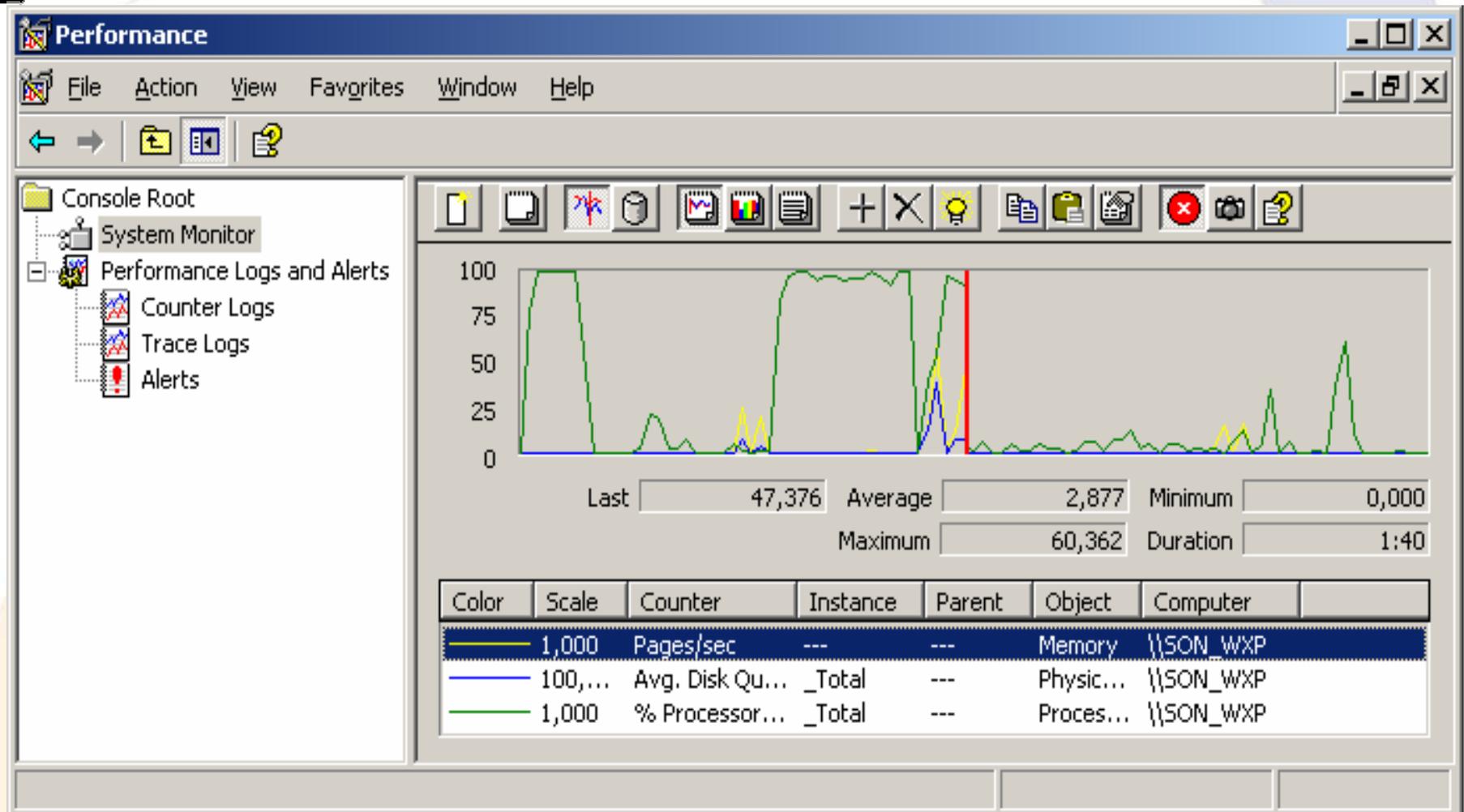


Entfernte Überwachung einer Anzahl von Computern

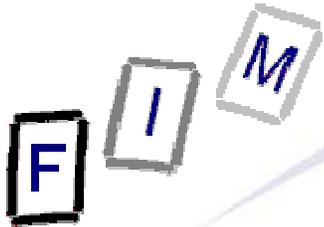




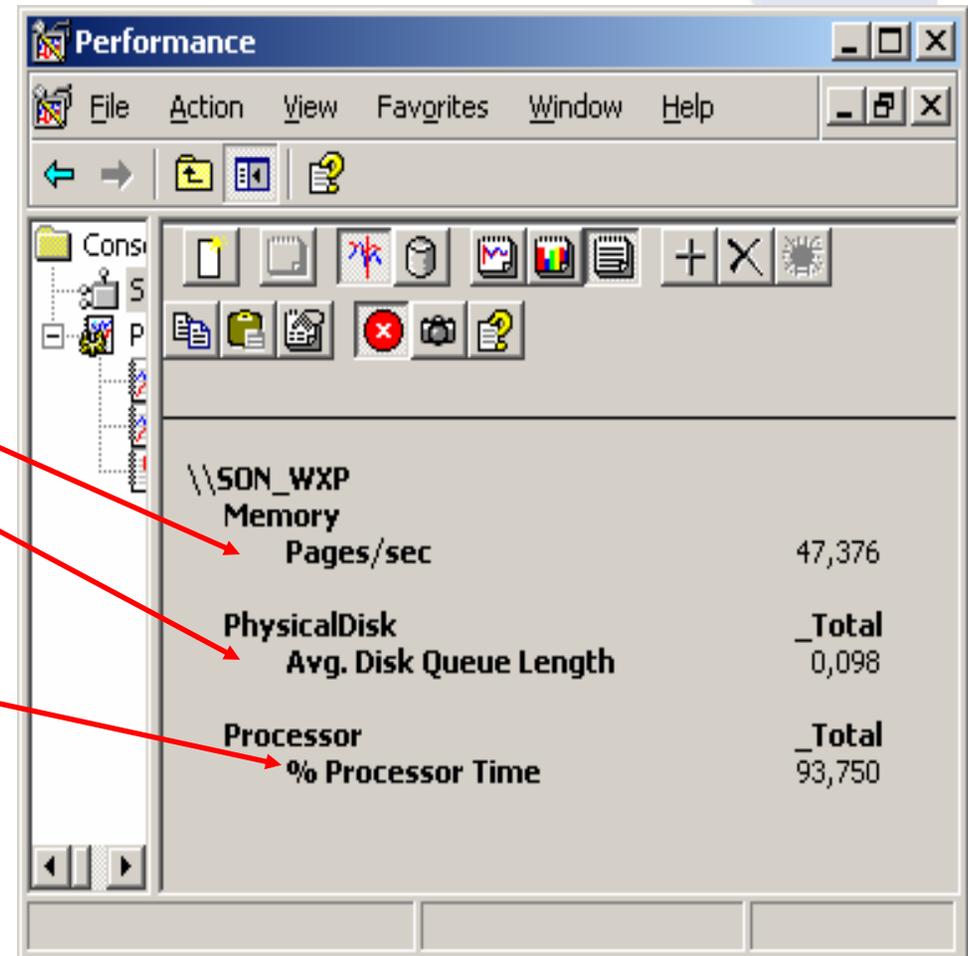
Graphische Darstellung



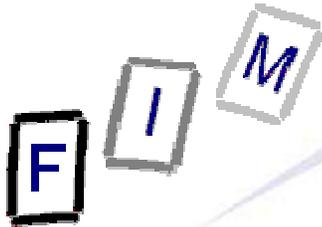
Anomalien für Menschen leicht erkennbar,
für Computer oft praktisch unmöglich!



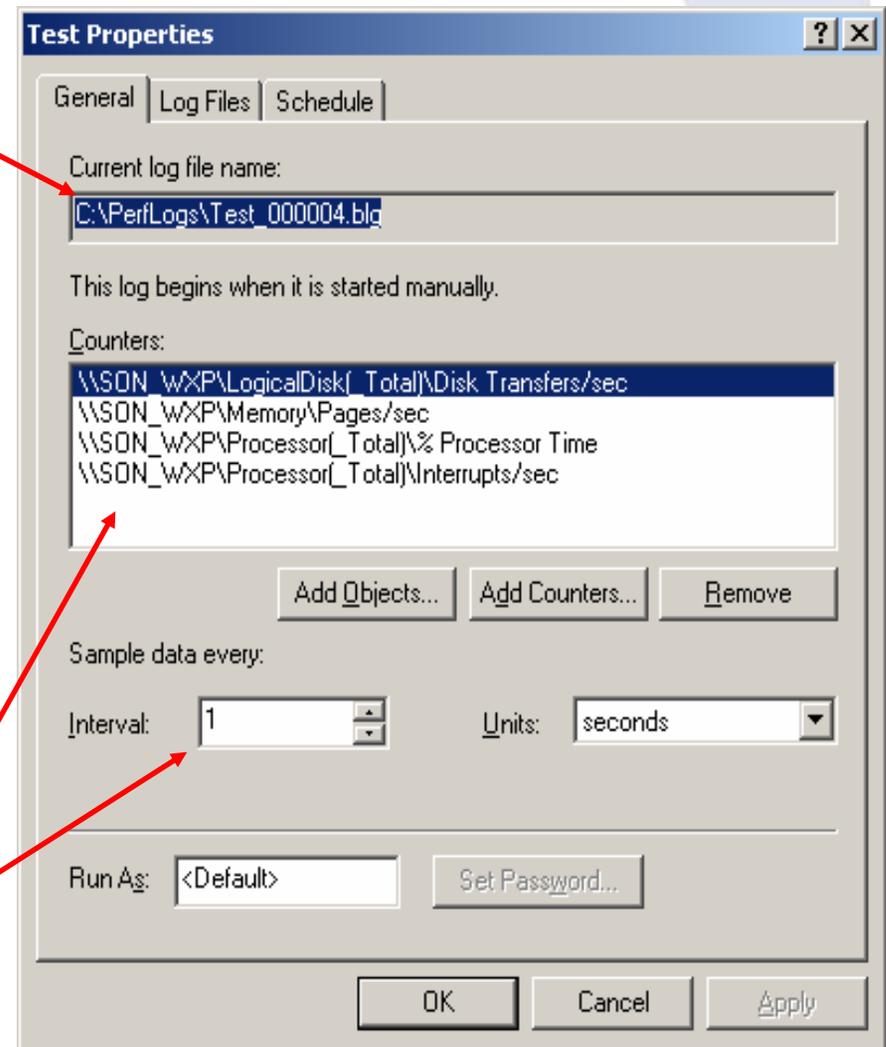
- Einfache Darstellung als Tabelle
 - Konfigurierbar
- Werte:
 - Seitenaus- und einlagerungen
 - Durchschnitt der Lese- und Schreiboperationen die auf Ausführung warten
 - Prozessor Auslastung
 - » Non-idle time Anteil



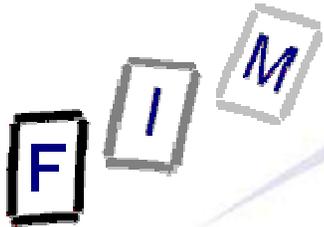
Aufzeichnung ohne (sofortige) Darstellung



- Derzeitiger Logfile-Name
 - Logs werden "rotiert", d. h. immer neu angelegt und nicht überschrieben
 - Echte Rotation: z. B. 5 Logfiles und diese werden "im Kreis" beschrieben
 - Ermöglicht "Blick zurück", ohne den Platzbedarf explodieren zu lassen
 - » Selten (z.B. DHCP): Wöchentlich, 4-5 Wochen
 - » Oft (z.B. HTTP): Täglich, 30 Tage oder länger
- Was überwacht wird
- Beobachtungsintervall



Aufzeichnung ohne (sofortige) Darstellung



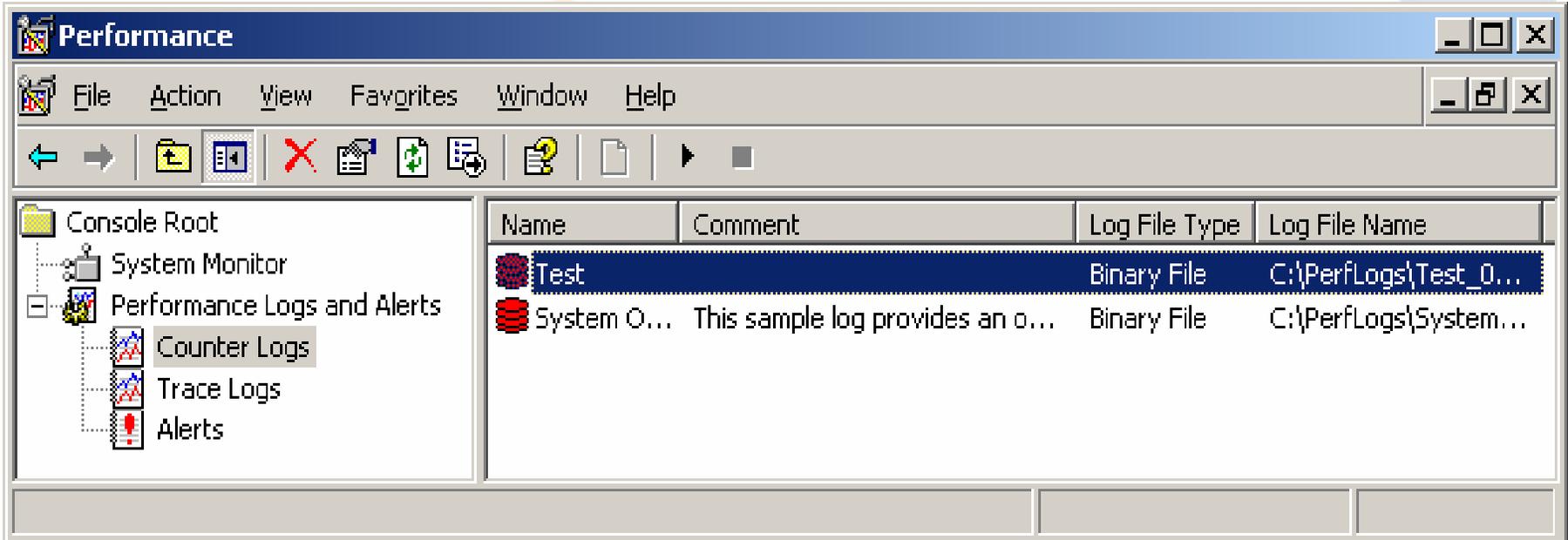
- Counter Logs:

- Regelmäßige Überwachung normaler Aktivität
 - » "Alle x Minuten"
 - Zeitabhängig
- Statistiken
 - » Hauptsächlich für Grafiken
- Großer Platzverbrauch

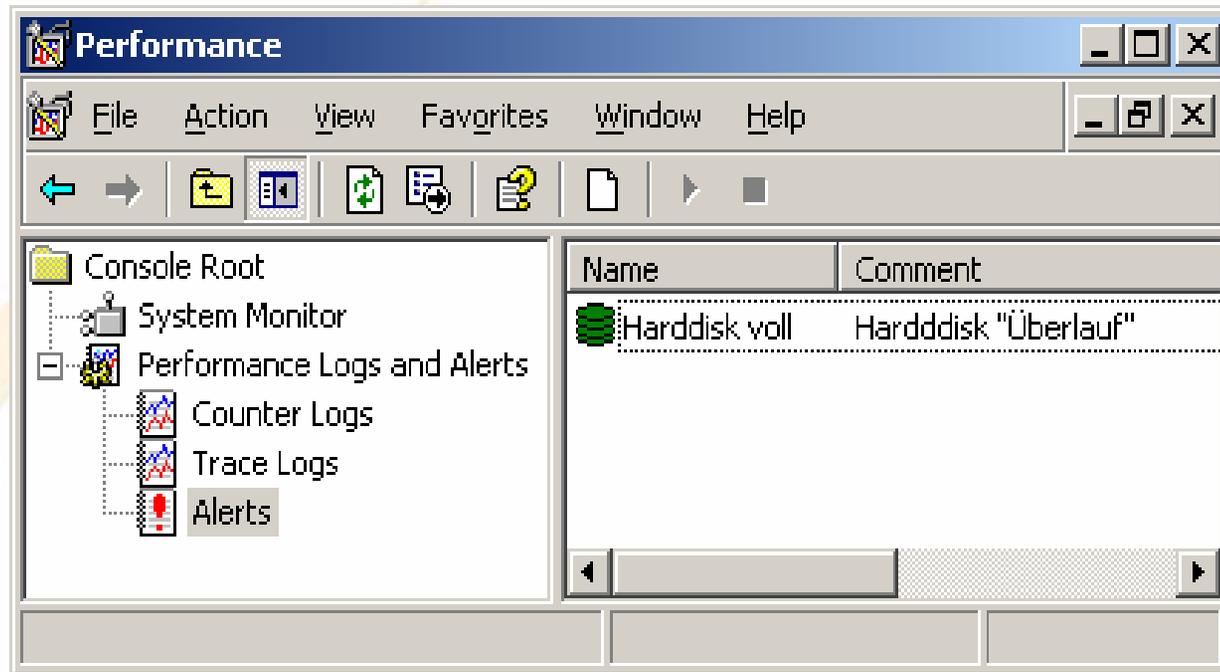
- Trace Logs:

- Ausnahmsweise Überwachung besonderer Aktionen
 - » "Bei Auftreten des Fehlers y"
 - Zeitunabhängig
- Ereignisse
- Unregelmäßig, nur bei Bedarf
- Platzverbrauch meist gering
 - » Falsche Konfiguration (=häufige Ereignisse) → Ev. riesige Größe!

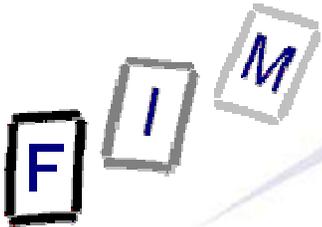
Aufzeichnung ohne (sofortige) Darstellung



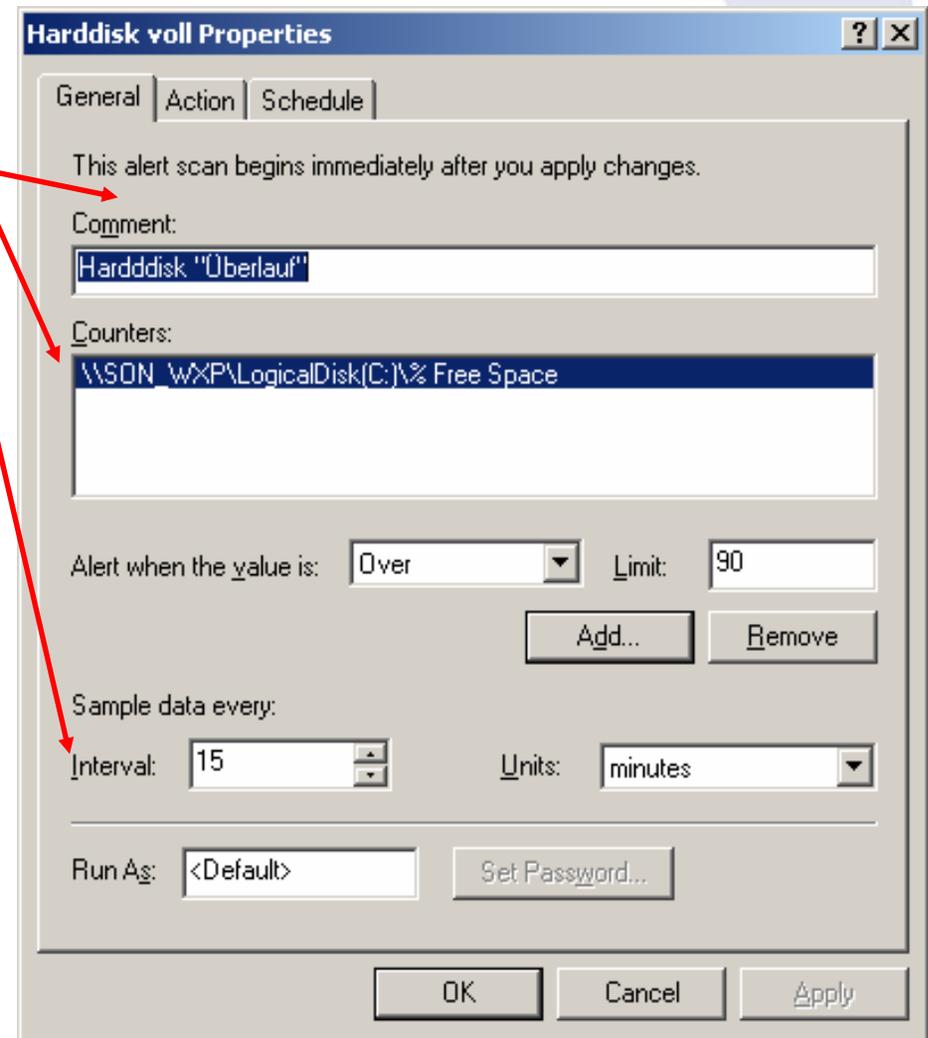
- Unter "Counter Logs" können Hintergrund-Überwachungen angelegt werden
 - Manueller oder automatischer Start möglich
 - Rechts die Liste aller aktuellen Konfigurationen
 - » Rot: Inaktiv
 - » Grün: Wird derzeit überwacht



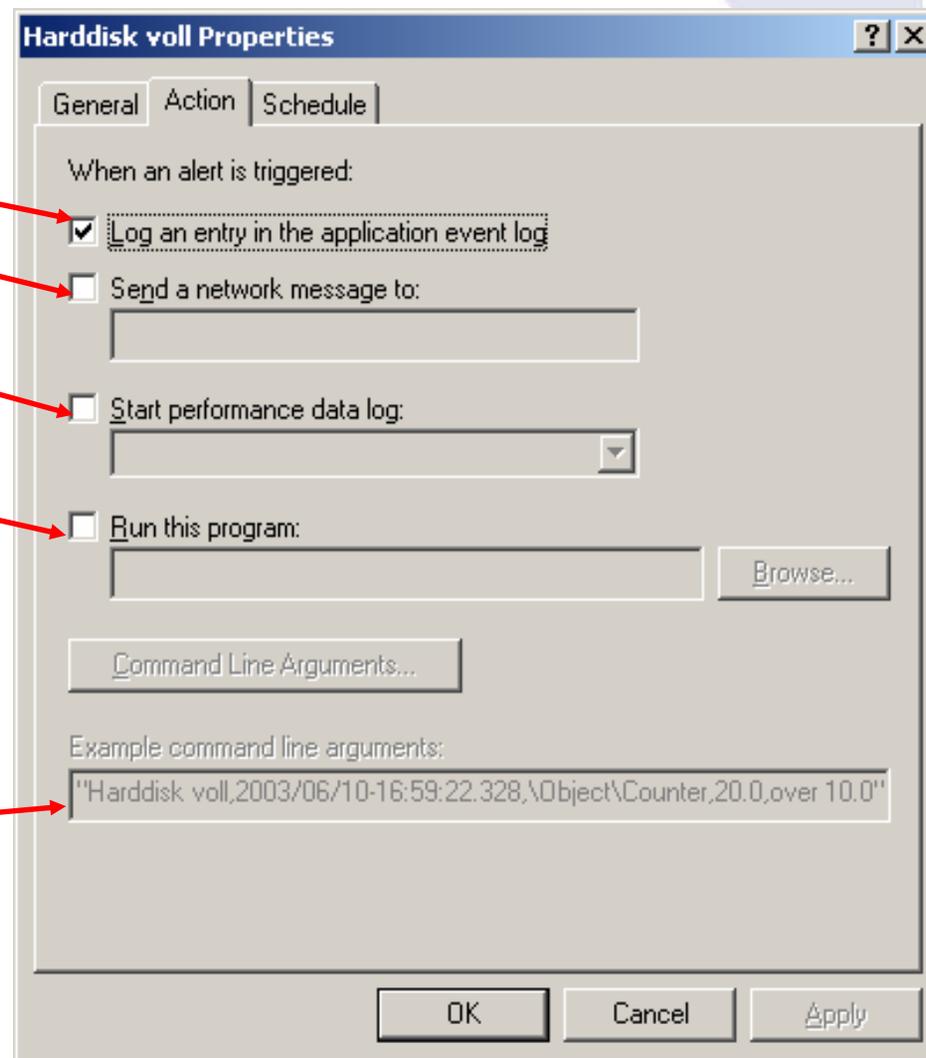
- Unter "Alerts" können Alarmer angelegt werden
 - Manueller oder automatischer Start (Standard) möglich
 - Rechts die Liste aller aktuellen Konfigurationen
 - » Rot: Inaktiv (Bei Auftreten keine Reaktion)
 - » Grün: Derzeit (=Sobald das Ereignis auftritt) aktiv

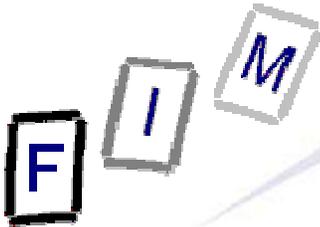


- Name des Alarms
- Was überwacht wird
- Wie oft überwacht wird
 - Überschreitungen dazwischen gehen verloren!
 - » Diese werden vollständig unterdrückt, daher genau wählen!
- Beispiel:
 - Wenn nur mehr 10 % der Festplatte C frei ist
 - » Dies kann sehr selten geprüft werden (z.B. 1x pro Tag würde ev. auch ausreichen)!



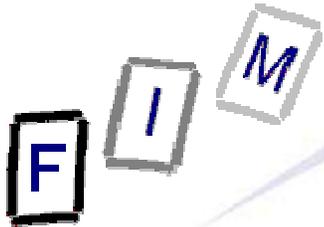
- Ereignis-Log
- Nachricht schicken
→ Netbios message
- Genauere Überwachung starten
- Externes Programm starten
→ Z.B. Temporäre Dateien löschen, Prozesse stoppen, ...
- Alarme können auch über Kommandozeile (=Batch-Programme) gemanagt werden: Entsprechende Argumente sind hier aufgelistet!





Linux Performance Monitoring

- Einheitliche Programme zum Sammeln und Darstellen rar
 - Beispiel: Complete System Performance Monitor for Linux by HP
 - » Open Source: <http://cspm.sourceforge.net/index.htm>
 - Alternative: Kommerzielle (relativ teure!) Programme
- Sonst:
 - Viele verschiedene (Kommandozeilen-)Programme
 - Direktes Auslesen (z.B. "/proc"-Dateisystem)
 - Auswertung von Log-Dateien
- Kein einheitliches Programm zur Darstellung
 - Verschiedenste Tools für konkrete Zwecke, oft aber auch für andere Eingangsdaten geeignet!
 - Beispiel: MRTG (Multi Router Traffic Grapher)
 - » <http://people.ee.ethz.ch/~oetiker/webtools/mrtg/>



Linux Performance Monitoring

- Daher oft (zusätzliche) Privatlösungen mit Scripts!
 - **Basieren meist auf Logfiles**
 - » Erfassungskomponente die in Logfiles schreibt
 - » Auswertungskomponente, welche Graphiken, Statistiken, etc. daraus erzeugt und/oder anzeigt
 - **Nachteil: Meistens keine Echtzeit-Anzeige möglich!**
- Alarme: Müssen meist händisch codiert werden