Arbeitsauftrag AAB03: Betriebssystemressourcen in C-Programmen anfordern

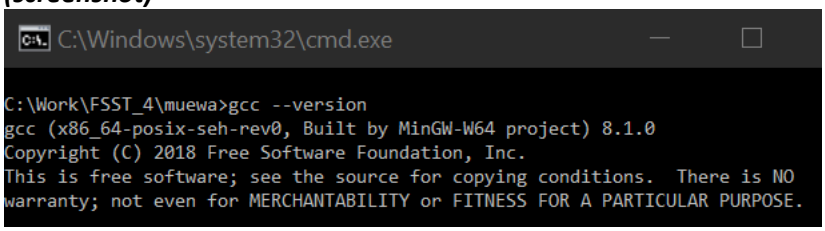
Datum: 07.02.2020

Autor: Mitterhuber Lorenz

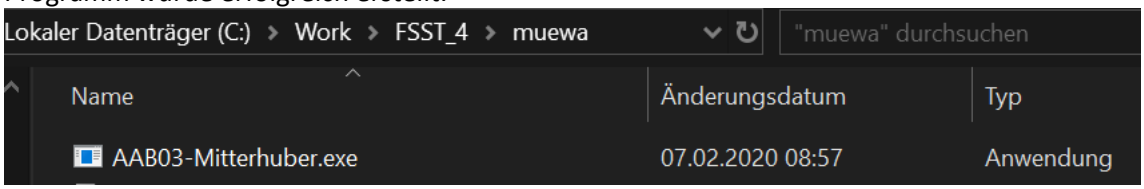
Hilfe: zum Arbeitsauftrag gehörige PPT-Folien

Bearbeite/beantworte/dokumentiere folgende Fragen
mit screenshots

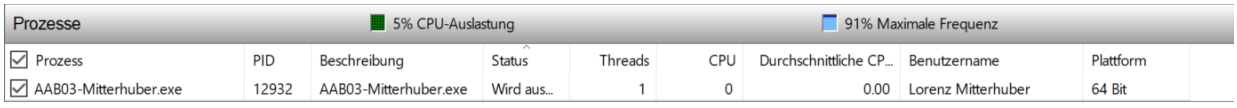
1. ***Installiert den mingw-w64 Compiler unter c:\programme***
   und öffnet mit dem script mingw-w64.bat eine CMD Box, die den Pfad bereits um den 64-Bit-Compiler erweitert hat.
   Testet ob der richtige Compiler gstartet wird:
   `gcc --version`
   (screenshot)



2. ***Übersetzt das Programm mit dem gcc Compiler.***
   `gcc -lpthread -o AAB03-Name.exe AAB03-os.c`
   (siehe Folien)
   Programm wurde erfolgreich erstellt.



3. ***Startet das erzeugte Programm in der CMD Box.***
   mingw-w64.bat starten und in diesem Terminal einfach den Name der .exe eingeben
   Selektiert das Programm AAB03-Name.exe .
   CPU:



   Welche PID hat das Programm?
   12932
   Wie viele Threads laufen, wie viel CPU benötigt das Programm?
   Thread: 1
   Momentane Auslastung: 0%
   Welche Module (Dateien und Bibliotheken) werden vom Programm verwendet?

   Die Endung .dll steht für Dynamic Link Library. Die Funktionalität von Programmen, die Sie auf einem dieser Windows-Betriebssysteme ausführen, wird eventuell ebenfalls über DLLs bereitgestellt. Programme können viele verschiedene Module beinhalten. Die einzelnen Module sind in DLLs enthalten und werden über diese verteilt.

Handelt es sich um eine 32bit oder eine 64bit Anwendung? (Spalten im Tab CPU erweitern)
64Bit Anwendung (siehe oberen Screenshot)

Memory
Wie viel Speicher wird zugesichert/verwendet?
RM

Prozess	PID	Harte Fehler/s	Zugesichert (KB)	Arbeitssatz (...	Freigabe mö...	Privat (KB)
AAB03-Mitterhuber.exe	10392	0	500	2.032	1.740	292

Dem Prozess „AAB03-Mitterhuber.exe“ werden 500KB zugesichert und davon werden 2.032KB verwendet. Ich wusste zuerst nicht welche Spalte anzeigt, wie viel Speicher verwendet wird. Deshalb habe ich den Prozess im Taskmanager analysiert. Die Spalte „Privat“ hat denselben Wert wie die Spalte „Arbeitsspeicher“ im Taskmanager

4. ***Reserviert mit dem c Kommando des Programms (calloc) ein Gigabyte Speicher.***

```
commands:
 c 500      reserve 500 1M blocks with calloc
 f          free last reserved buffer
 w 1000     write 1000 times at random pos to buffer
 l 10       loop cpu with 10 times 1 mio x=sin(x)
 t 5        loop cpu with 5 threads, each 100 times 1 mio x=sin(x)
 e          end program
enter command:c 1000
 1000 1M Bloecke reserviert ab Adresse: 0000000000904040
                          bis Adresse: 000000003F104040
```

Wie geht das?
Im Terminal den Befehl c und den gewünschten Speicher (in MB) eingeben, Das bedeutet, dass calloc aufgerufen wird und bei „c 1000“ 1GB reserviert wird.
Von welcher Adresse bis zu welcher Adresse liegt der Speicherbereich?
Der Speicherbereich geht von 0000000000904040 bis 000000003F104040
Kontrolliere die Größe mit einem Hex-Rechner
Anfangsadresse in Dezimal: 9.453.632
Endadresse in Dezimal: 1.058.029.632
Differenz: 1.048.576.000

Dokumentiert das Verhalten im Ressourcenmonitor.

Prozess	PID	Harte Fehler/s	Zugesichert (KB)	Arbeitssatz (KB)	Freigabe mö...	Privat (KB)
AAB03-Mitterhuber.exe	10392	0	1.026.504	2.044	1.748	296

Wird der Speicher bereits verwendet?
Nein wird nicht verwendet da noch keine Befehel ausgeführt wurden, deshalb wird der Speicher auch nicht beschrieben.
Wieviel Speicher ist zugesichert, wie viel in Verwendung?
Dem Prozess „AAB03-Mitterhuber.exe“ werden 1.026.504KB zugesichert und davon werden 2.044KB verwendet.

5. ***Belegt zufällige Teile des Speichers mit Daten (w Befehl)***

```
commands:
 c 500      reserve 500 1M blocks with calloc
 f          free last reserved buffer
 w 1000     write 1000 times at random pos to buffer
 l 10       loop cpu with 10 times 1 mio x=sin(x)
 t 5        loop cpu with 5 threads, each 100 times 1 mio x=sin(x)
 e          end program
enter command:w 500
fill 500 times a random position with 1MByte of 0xff
```

Was passiert im Ressourcenmonitor (Arbeitsspeicher)?

Prozess	PID	Harte Fehler/s	Zugesichert (KB)	Arbeitssatz (KB)	Freigabe mö...	Privat (KB)
AAB03-Mitterhuber.exe	10392	0	1.026.504	402.360	1.748	400.612

Der Arbeitssatz & Privat steigen stark an.

Fülle zufällig weitere Speicherbereiche.

Prozess	PID	Harte Fehler/s	Zugesichert (KB)	Arbeitssatz (KB)	Freigabe mö...	Privat (KB)
AAB03-Mitterhuber.exe	10392	0	1.026.504	638.872	1.748	637.124

Was passiert?

Der Arbeitssatz & Privat steigen weiter an.

Gib den zuletzt reservierten Speicher mit dem f Befehl (free) wieder frei.

Dokumentiert das Verhalten im Ressourcenmonitor mittels screenshot.

Prozess	PID	Harte Fehler/s	Zugesichert (KB)	Arbeitssatz (KB)	Freigabe mö...	Privat (KB)
AAB03-Mitterhuber.exe	10392	0	500	2.040	1.748	292

Zugesicherter Speicher wurde wieder frei gegeben!

6. Alloziert und beschreibt nun mehr als 4GB Speicher.

```
commands:
 c 500       reserve 500 1M blocks with calloc
 f           free last reserved buffer
 w 1000      write 1000 times at random pos to buffer
 l 10        loop cpu with 10 times 1 mio x=sin(x)
 t 5         loop cpu with 5 threads, each 100 times 1 mio x=sin(x)
 e           end program
enter command:c 4500
 4500 1M Bloecke reserviert ab Adresse: 000000007FFF2040
                        bis Adresse: 00000001993F2040

commands:
 c 500       reserve 500 1M blocks with calloc
 f           free last reserved buffer
 w 1000      write 1000 times at random pos to buffer
 l 10        loop cpu with 10 times 1 mio x=sin(x)
 t 5         loop cpu with 5 threads, each 100 times 1 mio x=sin(x)
 e           end program
enter command:w 4500
fill 4500 times a random position with 1MByte of 0xff
```

Prozess	PID	Harte Fehler/s	Zugesichert (KB)	Arbeitssatz (KB)	Freigabe mö...	Privat (KB)
AAB03-Mitterhuber.exe	10392	0	4.617.536	2.918.568	1.748	2.916.820

Es werden nicht die kompletten 4,5GB beschrieben, da das Programm mit w 4500 an zufällige Stellen des Speichers schreibt und somit es nicht möglich ist, dass der ganze Speicher voll wird. Den es können auch Stellen überschrieben („doppelt getroffen“) werden.

Wäre das mit/in einer 32-Bit Anwendung möglich?

Nein, das ist nicht möglich, denn bei 32-Bit Anwendungen liegt die Grenze bei 4GB → 2^{32}= 4.294.967.296

7. Starte eine weitere Instanz des Programms in der CMD-Box:

Prozess	PID	Harte Fehler/s	Zugesichert (KB)	Arbeitssatz (KB)	Freigabe mö...	Privat (KB)
AAB03-Mitterhuber.exe	10392	0	4.617.536	2.918.416	1.748	2.916.668
AAB03-Mitterhuber.exe	9228	0	472	1.964	1.688	276

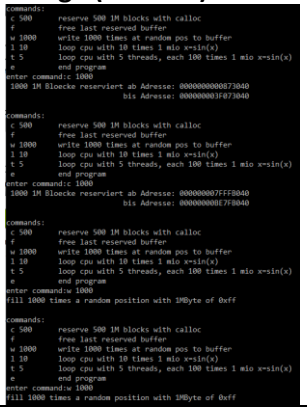
Was erkennt man im Ressourcenmonitor?

Man erkennt, dass ein weiterer Prozess mit der gleichen Bezeichnung im RM aufscheint.

Wie unterscheiden sich die beiden Prozesse?

Die Prozesse unterscheiden sich in den Bereichen zugsicherten Speicher und Arbeitssatz.
Außerdem ist logischerweise auch die PID unterschiedlich.

8. Belege auch in der zweiten Programminstanz Speicherplatz.
Belege (allocate) dort mehrmals 1GByte und beschreibe den Speicher.



Hier wird 2 mal 1GB reserviert und 2 mal 1000 mal beschrieben

Welcher Speicherbereich wird nun verwendet (Anfangs- und Endadressen)?
2. Prozess
1. Reservierung: Startadresse: 0000000000873040 → Dezimal: 8859712
Endadresse: 000000003F073040 → Dezimal: 1057435712
2. Reservierung: Startadresse: 000000007FFFB040 → Dezimal: 2147463232
Endadresse: 00000000BE7FB040 → Dezimal: 3196039232

Überschneidet sich der Speicherbereich tatsächlich mit dem anderen Prozess?
Gib eine Erklärung dazu!
1. Prozess
Startadresse: 000000007FFF2040 → Dezimal: 2147426368
Endadresse: 00000001993F2040 → Dezimal: 6866018368
Erklärung: Jeder Prozess hat seinen logischen Adressraum, dieser kann sich mit den logischen Adressräumen anderer Prozesse überschneiden. Die physischen Adressräume können sich nicht überschneiden.

Wie groß ist der Adressraum in einem 32 bit-Betriebssystem?
4GB

Wie wird das dem physisch im Rechner eingebauten Memory zugeordnet?
Die MMU (Memory Management Unit), ein in die CPU integrierter Hardware-Baustein, ist dafür zuständig, dass der logische Adressraum dem Memory ohne Überschneidungen zugeordnet.

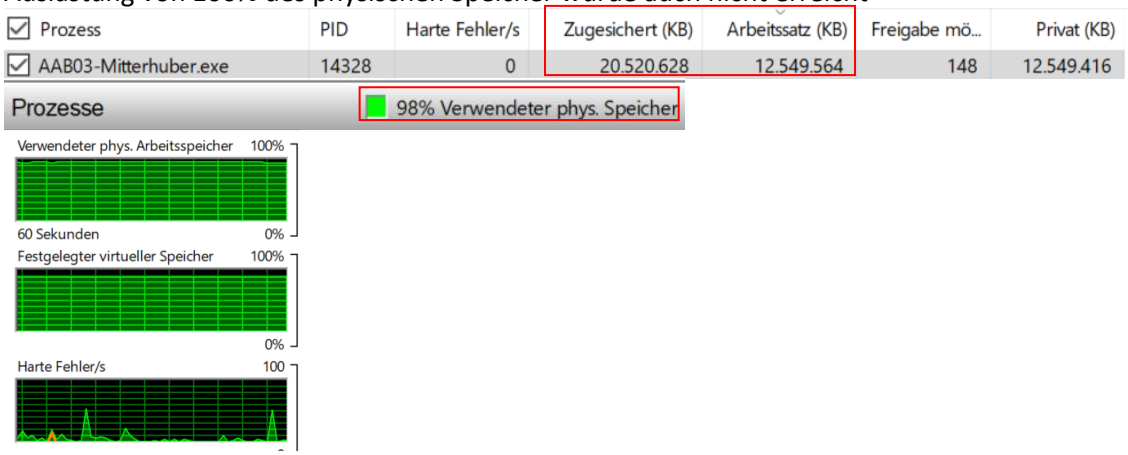
Wie viel Speicher kannst Du mit einem Prozess in deinem Betriebssystem anfordern.
(der retournierte Pointer buffer ist ungleich 0x00)

```
commands:
 c 500      reserve 500 1M blocks with calloc
 f          free last reserved buffer
 w 1000     write 1000 times at random pos to buffer
 l 10       loop cpu with 10 times 1 mio x=sin(x)
 t 5        loop cpu with 5 threads, each 100 times 1 mio x=sin(x)
 e          end program
enter command:c 60000
60000 1M Bloecke reserviert ab Adresse: 0000000000000000
                        bis Adresse: 0000000EA6000000
```

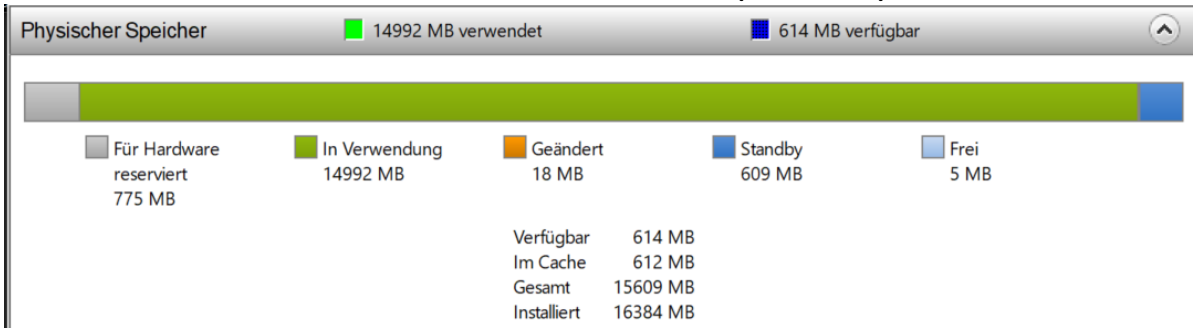
```
commands:
 c 500      reserve 500 1M blocks with calloc
 f          free last reserved buffer
 w 1000     write 1000 times at random pos to buffer
 l 10       loop cpu with 10 times 1 mio x=sin(x)
 t 5        loop cpu with 5 threads, each 100 times 1 mio x=sin(x)
 e          end program
enter command:c 59000
59000 1M Bloecke reserviert ab Adresse: 0000000F26005040
                        bis Adresse: 0000001D8D805040
```

Es können Maximal 59GB alloziert werden, bei 60GB gibt das Programm eine Adresse mit aussschleßlich 0 zurück.

9. **Was passiert wenn der physisch vorhandenen Speicher auf dem Rechner von den Prozessen aufgebraucht wird?**
Für diesen Punkt wurde einem Prozess mindestens 20GB zugesichert (mein Rechner besitzt 16GB RAM) Anschließend wurde der Speicher mit dem Befehl w 20000 beschrieben, allerdings belegte der Prozess höchstens 12,6GB, da bei diesem Wert der physische Speicher laut Ressourcenmonitor zu 98% belegt war. Daher versuchte ich erneut den Speicher zu beschreiben, doch die Werte blieben annähernd gleich und eine Auslastung von 100% des physischen Speicher wurde auch nicht erreicht

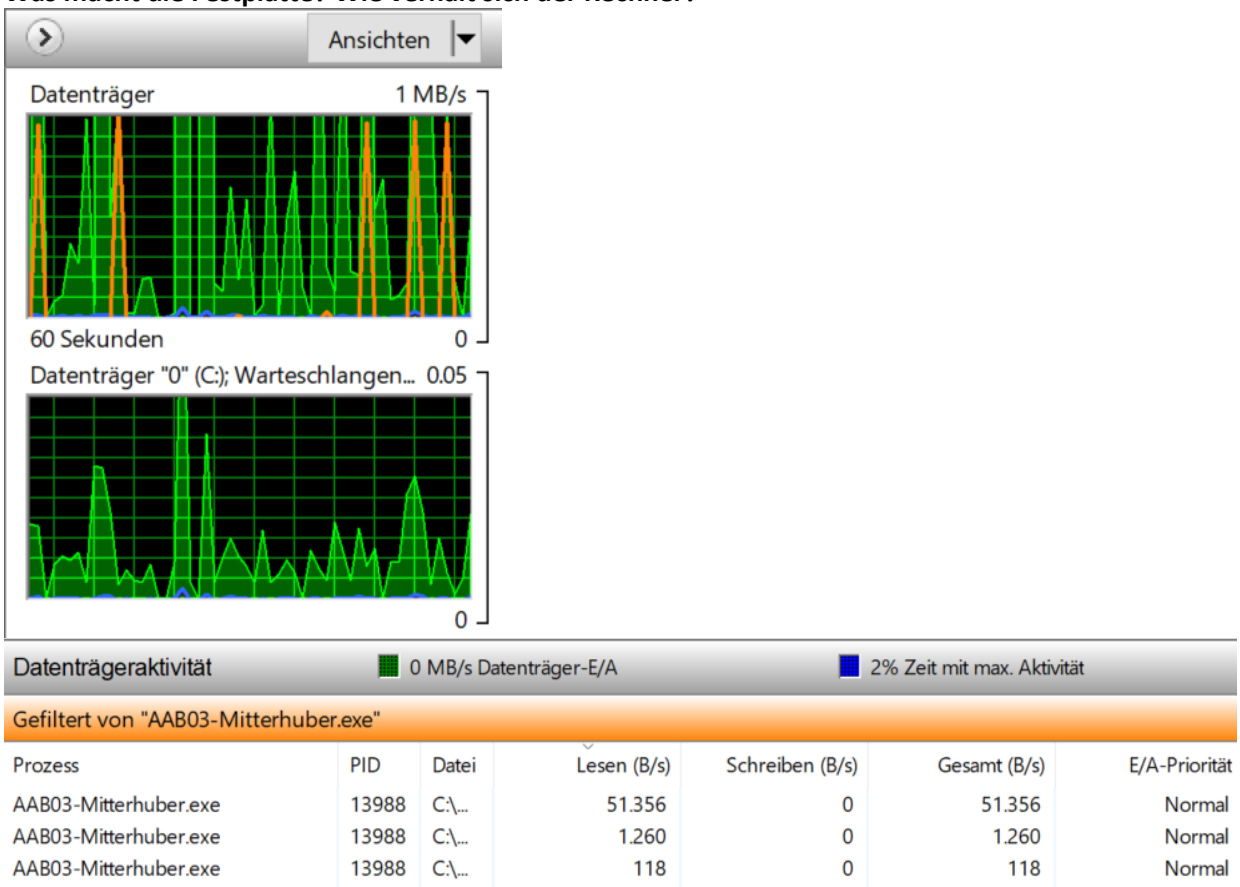


Wie sieht der Übersichtsbalken im Ressourcenmonitor aus? (screenshot)



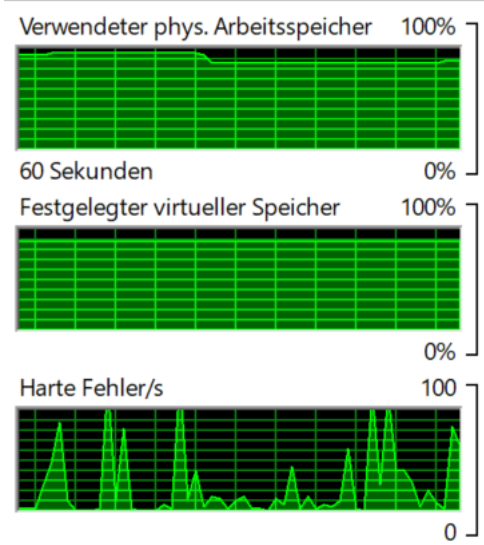
Beim Beschreiben des Speichers des Prozesses stieg der grüne Balken („In Verwendung“) stark an und der blaue („Standby“) wurde immer kleiner, allerdings verschwand der blaue nie komplett.

Was macht die Festplatte? Wie verhält sich der Rechner?



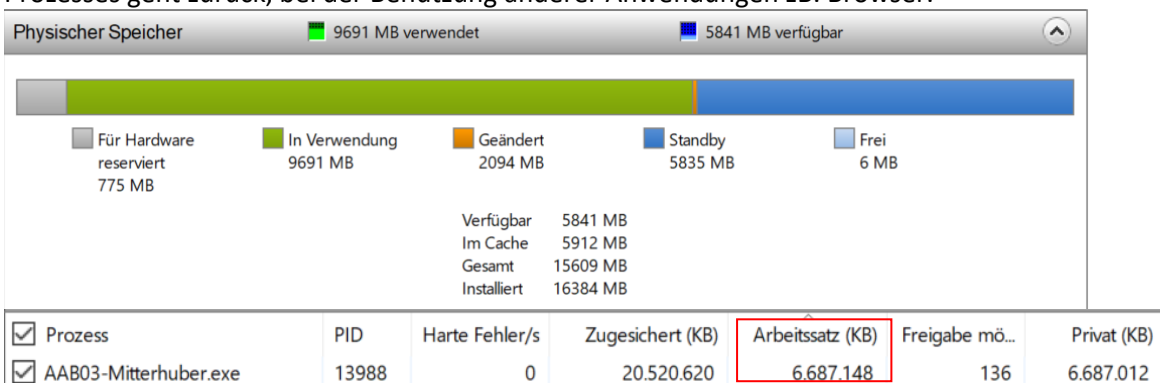
Prozess	PID	Datei	Lesen (B/s)	Schreiben (B/s)	Gesamt (B/s)	E/A-Priorität
AAB03-Mitterhuber.exe	13988	C:\...	51.356	0	51.356	Normal
AAB03-Mitterhuber.exe	13988	C:\...	1.260	0	1.260	Normal
AAB03-Mitterhuber.exe	13988	C:\...	118	0	118	Normal

Was passiert, wenn Du zu anderen Anwendungen am Rechner umschaltest?



Beim Umschalten zu anderen Anwendungen bleibt der verwendete physische Arbeitsspeicher annähernd gleich (minimaler Abfall), jedoch im Diagramm der Harten Fehler gibt es deutliche Änderungen.

Nach längerer Zeit (ca. 5min) vergrößert sich der Standby Balken in der Übersicht und der Arbeitssatz des Prozesses geht zurück, bei der Benutzung anderer Anwendungen zB. Browser.



10. Was passiert, wenn Du eine der beiden Programminstanzen beendest?
Wie sieht die Speicherbelegung im Ressourcenmonitor dann aus?(screenshot)

Im Diagramm des Verwendeten physischen Arbeitsspeicher wird ersichtlich, dass der Pegel stark sinkt. In der Übersicht kann man erkennen, dass nun wieder viel Speicher frei geworden ist.

11. Starte das Programm erneut und verwende die Funktion l mit 100 Durchläufen.

```
commands:
 c 500      reserve 500 1M blocks with calloc
 f          free last reserved buffer
 w 1000     write 1000 times at random pos to buffer
 l 10       loop cpu with 10 times 1 mio x=sin(x)
 t 5        loop cpu with 5 threads, each 100 times 1 mio x=sin(x)
 e          end program
enter command:l 100
cpu loop 100 Mio times x=sin(x)
... cpu loop 100 Mio times x=sin(x)
```

Was beobachtest Du im Ressourcenmonitor?
(screenshot)

Vorher:

CPU

Prozess	PID	Beschreibung	Status	Threads	CPU	Durchschnittliche CP...	Benutzername	Plattform
AAB03-Mitterhuber.exe	12092	AAB03-Mitterhuber.exe	Wird aus...	1	0	0.00	Lorenz Mitterhuber	64 Bit

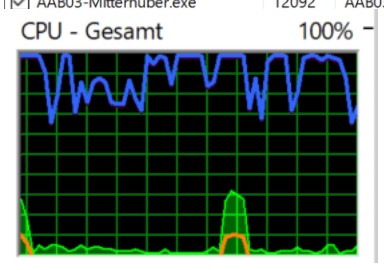
Arbeitsspeicher

Prozess	PID	Harte Fehler/s	Zugesichert (KB)	Arbeitssatz (KB)	Freigabe mö...	Privat (KB)
AAB03-Mitterhuber.exe	12092	0	476	1.940	1.732	208

Nachher:

CPU

Prozess	PID	Beschreibung	Status	Threads	CPU	Durchschnittliche CP...	Benutzername	Plattform
AAB03-Mitterhuber.exe	12092	AAB03-Mitterhuber.exe	Wird aus...	1	0	0.79	Lorenz Mitterhuber	64 Bit



Arbeitsspeicher

Prozess	PID	Harte Fehler/s	Zugesichert (KB)	Arbeitssatz (KB)	Freigabe mö...	Privat (KB)
AAB03-Mitterhuber.exe	12092	0	476	1.948	1.740	208

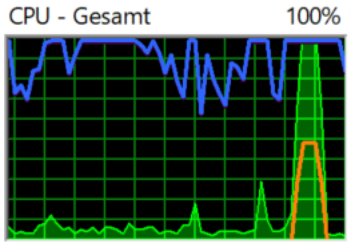
Der Vergleich Vorher-Nachher zeigt, dass sich nur die durchschnittliche CPU-Auslastung zugenommen hat. Im Bereich des Arbeitsspeichers gibts es keine wirklichen Unterschiede.

12. **Verwende die Funktion t (für mehrere Threads starten) mit 4 parallelen Threads:**
Was beobachtest Du?
Es wirkt sich nicht wirklich auf die Geschwindigkeit anderer Programme aus, trotz der hohen CPU Auslastung.
Wie sieht das im Ressourcenmonitor aus?

Prozesse — 110% CPU-Auslastung — 205% Maximale Frequenz

Prozess	PID	Beschreibung	Status	Threads	CPU	Durchschnittliche CP...	Benutzername
AAB03-Mitterhuber.exe	6492	AAB03-Mitterhuber.exe	Wird aus...	5	50	4.64	Lorenz Mitterhuber

Hier sieht man, dass die Anzahl der Threads auf 5 angestiegen ist (vorher betrug der Wert 1) und die CPU-Auslastung liegt nun bei über 100%.
Dokumentiere auch den Verlauf der CPU-Nutzung (screenshot)



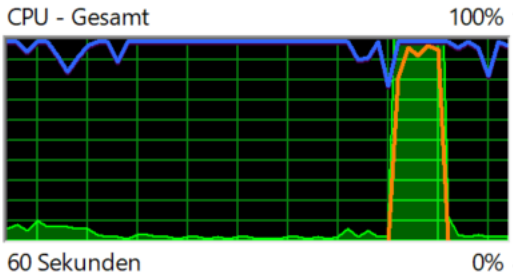
Hier sieht man, dass die Auslastung der CPU für kurze Zeit stark ansteigt

13. **Was passiert, wenn Du 8 Threads oder mehr startest?**
Wie verhält sich der Rechner?

Prozesse — 205% CPU-Auslastung — 204% Maximale Frequenz

Prozess	PID	Beschreibung	Status	Threads	CPU	Durchschnittliche...	Benutzername	Plattform
AAB03-Mitterhuber.exe	6492	AAB03-Mitterhuber.exe	Wird aus...	9	98	6.15	Lorenz Mitterhuber	64 Bit

Der Rechner wird nicht kaum langsamer, trotz der hohen Auslastung.
Laufen die Threads bei 4 Hyperthreading Kernen tatsächlich parallel?
Ja, die Threads laufen parallel. Die Idee von Hyperthreading ist nämlich, dass sich beispielsweise 2 Threads die Ressourcen teilen, welche für einen vollständigen Kern notwendig wären. So kann ein Thread die Ressourcen verwenden, die der andere momentan nicht benötigt.
Wie wirkt sich das Starten von mehreren Threads auf die Laufzeit eines Threads aus?
(screenshot der CPU-Nutzung)



Ja die Laufzeit verlängert sich.

14. **Übersetzt das Programm auch in der virtuellen Linux-Mate-Maschine.**
    Vorsicht: Hier lautet der gcc Aufruf wie folgt: (Libraries nach dem C-File anfügen)
    gcc –o AAB03-walter AAB03-OSfromC.c –lpthread –lm

```
lorenz@lorenz-virtual-machine:~/Schreibtisch/AAB03$ gcc -o AAB03-Mitterhuber AAB
03-os.c -lpthread -lm
lorenz@lorenz-virtual-machine:~/Schreibtisch/AAB03$ ./AAB03-Mitterhuber

commands:
 c 500      reserve 500 1M blocks with calloc
 f          free last reserved buffer
 w 1000     write 1000 times at random pos to buffer
 l 10       loop cpu with 10 times 1 mio x=sin(x)
 t 5        loop cpu with 5 threads, each 100 times 1 mio x=sin(x)
 e          end program
enter command:
```

Das C-File wurde im Ordner „AAB03“, welcher am Schreibtisch liegt, gespeichert. Daher musste zuerst der Pfad gewechselt werden (cd-Commands) und dann wurde der vorgegebene Befehl eingegeben.
Mit ./Name_der_exe wird das Programm ausgeführt (siehe Screenshot)

15. **Spielt Euch auch unter Linux mit dem Programm und beobachtet das Ergebnis mit top und dem Ressourcenmonitor.**

1GB Speicher reserviert und 1000mal an einer zufälligen Stelle beschrieben:

```
commands:
 c 500      reserve 500 1M blocks with calloc
 f          free last reserved buffer
 w 1000     write 1000 times at random pos to buffer
 l 10       loop cpu with 10 times 1 mio x=sin(x)
 t 5        loop cpu with 5 threads, each 100 times 1 mio x=sin(x)
 e          end program
enter command:c 1000
 1000 1M Bloecke reserviert ab Adresse: 0x7f8478b90010
                          bis Adresse: 0x7f84b7390010

commands:
 c 500      reserve 500 1M blocks with calloc
 f          free last reserved buffer
 w 1000     write 1000 times at random pos to buffer
 l 10       loop cpu with 10 times 1 mio x=sin(x)
 t 5        loop cpu with 5 threads, each 100 times 1 mio x=sin(x)
 e          end program
enter command:w 1000
fill 1000 times a random position with 1MByte of 0xff
```

Prozesseigenschaften von »AAB03-Mitterhuber« (PID 2884):

Prozessname	AAB03-Mitterhuber
Benutzer	lorenz (1000)
Status	Schläft
Arbeitsspeicher	641,8 MiB
Virtueller Speicher	1010,2 MiB
Nicht auslagerbarer Speicher	641,8 MiB
Schreibbarer Speicher	n.v.
Gemeinsamer Speicher	1,9 MiB
X-Server-Speicher	n.v.
CPU	0%
CPU-Zeit	0:02.86
Gestartet	Heute 20:38
Nice-Wert	0
Priorität	Normal
Kennung	2884
Sicherheitskontext	unconfined
Befehlszeile	./AAB03-Mitterhuber

Arbeitsspeicher benutzt und reserviert

1000mal 1Mio mal sin(x) – „l 1000“:

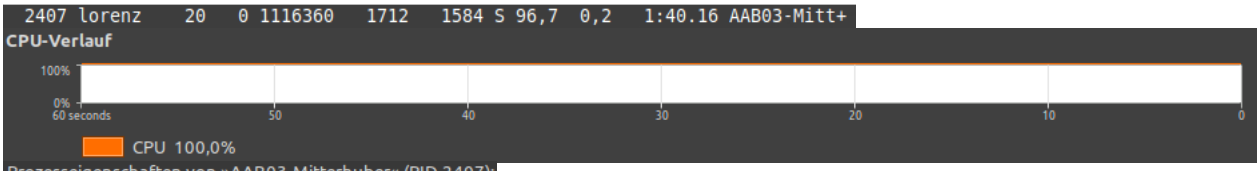
AAB03-Mitterhuber	Läuft	68	0	2407	1,7 MiB	0

Hier wird ersichtlich, dass die CPU-Auslastung ansteigt.

Loop mit 10 threads:

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	ZEIT+	BEFEHL
2407	lorenz	20	0	1116360	1712	1584	S	96,7	0,2	1:40.16	AAB03-Mitt+

CPU-Verlauf



Prozesseigenschaften von »AAB03-Mitterhuber« (PID 2407):

Prozessname	AAB03-Mitterhuber
Benutzer	lorenz (1000)
Status	Läuft
Arbeitsspeicher	1,8 MiB
Virtueller Speicher	1,0 GiB
Nicht auslagerbarer Speicher	1,8 MiB
Schreibbarer Speicher	n.v.
Gemeinsamer Speicher	1,6 MiB
X-Server-Speicher	n.v.
CPU	89%
CPU-Zeit	1:59.68
Gestartet	Heute 21:16
Nice-Wert	0
Priorität	Normal
Kennung	2407
Sicherheitskontext	unconfined
Befehlszeile	./AAB03-Mitterhuber

Hier sieht man, dass die CPU seht lange zu fast 100% ausgelastet ist.

Der oberste Screenshot entspricht einem Auszug des Top-Befehls.

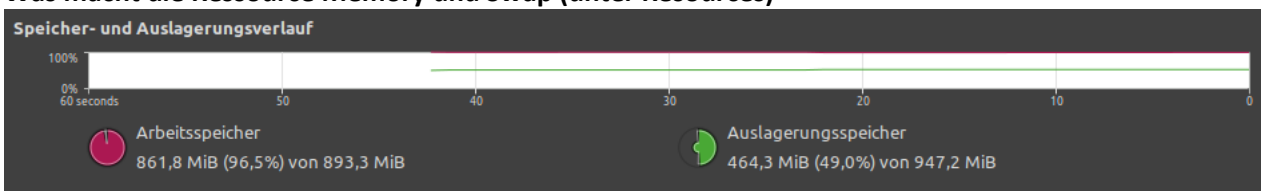
Könnt Ihr mehr Speicher anfordern, als Ihr der virtuellen Maschine gegönnt habt?

Prozesseigenschaften von »AAB03-Mitterhuber« (PID 3227):

Prozessname	AAB03-Mitterhuber
Benutzer	lorenz (1000)
Status	Schläft
Arbeitsspeicher	1,2 MiB
Virtueller Speicher	3,4 GiB
Nicht auslagerbarer Speicher	1,2 MiB
Schreibbarer Speicher	n.v.
Gemeinsamer Speicher	1,1 MiB
X-Server-Speicher	n.v.
CPU	0%
CPU-Zeit	0:00.00
Gestartet	Heute 21:04
Nice-Wert	0
Priorität	Normal
Kennung	3227

Hier wurden für den Prozess mit dem Befehl *„c 3500“* 3,5GB reserviert, sprich ja es ist möglich mehr Speicher anzufordern!

Was macht die Ressource Memory und Swap (unter Resources)



Ich habe dem Linux-OS 1GB RAM zugewiesen. Für diesen Versuch habe ich einem Prozess 2GB Speicher zugesichert und dann mit „w 1000“ beschrieben. Hier sieht man, dass bei vollem Arbeitsspeicher der Auslagerungsspeicher beschrieben wird.

Was ist und wo liegt der SWAP?

Der Begriff SWAP steht für Auslagerungsspeicher, dieser wird beschrieben, sobald der Arbeitsspeicher voll ist. Dieser Auslagerungsspeicher liegt auf der Festplatte, dort wo die VM angelegt wurde.

16. **Abgabe:**
    Gib den ausgefüllten Arbeitsauftrag mit den beantworteten Fragen und den screenshots als AAB03-*Name*.pdf im moodle ab!