

AAB03: OSfromC

Fabio Plunser
Betreuer : Walter Mueller

March 17, 2020

Inhaltsverzeichnis

1	Installiert den mingw-w64 Compile unter c:Programme und öffnet mit den script mingw-w64.bat eine CMD Box, die den Pfad bereits um den 64-Bit-Compiler erweitert hat	1
2	Übersetzt das Programm mit dem gcc Compiler.	1
3	Startet das erzeugte Programm in der CMD Box.	1
4	Reserviert mit dem c Kommando des Programms (calloc) ein Gigabyte Speicher.	3
4.1	Wie geht das?	3
4.2	Von welcher Adresse bis zu welcher Adresse liegt der Speicherbereich?	3
4.3	Kontrolliere die Größe mit einem Hex-Rechner	3
4.4	Dokumentiert das Verhalten im Ressourcenmonitor.	3
4.4.1	Wird der Speicher bereits verwendet?	3
4.4.2	Wieviel Speicher ist zugesichert, wie viel in Verwendung?	4
5	Belegt zufällige Teile des Speichers mit Daten (w Befehl)	4
5.1	Was passiert im Ressourcenmonitor (Arbeitsspeicher)?	4
5.2	Fülle zufällig weitere Speicherbereiche.	4
5.2.1	Was passiert?	4
6	Alloziert und beschreibt nun mehr als 4GB Speicher.	4
6.1	Wäre das mit/in einer 32-Bit Anwendung möglich?	5
7	Starte eine weitere Instanz des Programms in der CMD-Box:	5
7.1	Was erkennt man im Ressourcenmonitor?	5
7.2	Wie unterscheiden sich die beiden Prozesse?	5
8	Belege auch in der zweiten Programminstanz Speicherplatz.	5
8.1	Belege (allocate) dort mehrmals 1GByte und beschreibe den Speicher	5
8.2	Welcher Speicherbereich wird nun verwendet (Anfangs- und Endadressen)?	6
8.3	Überschneidet sich der Speicherbereich tatsächlich mit dem anderen Prozess?	6
8.4	Gib eine Erklärung dazu!	6
8.5	Wie groß ist der Adressraum in einem 32 bit-Betriebssystem?	6
8.6	Wie wird das dem physisch im Rechner eingebauten Memory zugeordnet?	6
8.7	Wie viel Speicher kannst Du mit einem Prozess in Deinem Betriebssystem anfordern.(der retournierte Pointer buffer ist ungleich 0x00)	7
9	Was passiert wenn der physisch vorhandenen Speicher auf dem Rechner von den Prozessen aufgebraucht wird?	8
9.1	Wie sieht der Übersichtsbalken im Ressourcenmonitor aus? (screenshot)	8
9.2	Was macht die Festplatte? Wie verhält sich der Rechner?	8
9.3	Was passiert, wenn Du zu anderen Anwendungen am Rechner umschaltest?	9
10	Was passiert, wenn Du eine der beiden Programminstanzen beendest?	9
10.1	Wie sieht die Speicherbelegung im Ressourcenmonitor dann aus?(screenshot)	9
11	Starte das Programm erneut und verwende die Funktion 1 mit 100 Durchläufen.	10
11.1	Was beobachtest Du im Ressourcenmonitor?(screenshot)	10

12 Verwende die Funktion t (für mehrere Threads starten) mit 4 parallelen Threads:	10
12.1 Was beobachtest Du?	10
12.2 Wie sieht das im Ressourcenmonitor aus?	10
12.3 Dokumentiere auch den Verlauf der CPU-Nutzung (Screenshot)	11
13 Was passiert, wenn Du 8 Threads oder mehr startest?	11
13.1 Wie verhält sich der Rechner?	11
13.2 Laufen die Threads bei 4 Hyperthreading Kernen tatsächlich parallel?	11
13.3 Wie wirkt sich das Starten von mehreren Threads auf die Laufzeit eines Threads aus?(Screenshot der CPU-Nutzung)	11
14 Übersetzt das Programm auch in der virtuellen Linux-Mate-Maschine.	12
15 Spielt Euch auch unter Linux mit dem Programm und beobachtet das Ergebnis mit top und dem Ressourcenmonitor.	12
15.1 Könnt Ihr mehr Speicher anfordern, als Ihr der virtuellen Maschine gegönnt habt?	12
15.2 Was macht die Ressource Memory und Swap (unter Resources)	13
15.3 Was ist und wo liegt der SWAP?	13

Abbildungsverzeichnis

1 gcc-version	1
2 Programm-Ressourcenmonitor	1
3 Programm-Handles	2
4 Programm-RAM	2
5 Programm-RAM-Reservierung	3
6 Byte in GB	3
7 RAM-Reservierung-RM	3
8 W-1000	4
9 W-4*1000	4
10 4.5GB-Rservierung	4
11 4.5GB-Verwendung	5
12 2-Instanzen	5
13 2-Instanzen-RAM-Verwendung	5
14 2-Instanzen-Speicherraum-1	6
15 2-Instanzen-Speicherraum-2	6
16 Maximal-reservierbarer-Speicher	7
17 Arbeitsspeicher-Übersichtsbalken	8
18 Datenträger-bei-beschreibung	8
19 Speicher-nach-Prozess-Beendung	9
20 Speicher-nach-Prozess-Beendung	9
21 l-100-1	10
22 4-Threads-Auslastung	10
23 4-Threads-Prozess	10
24 8-Threads-Auslastung	11
25 8-Threads-Prozess	11
26 8-Threads-Auslastung	11
27 Linux-Programm-Übersetzung	12
28 Linux-maximal-reservierbarer-Speicher	12
29 Linux-Memory-Swap	13

AAB02: OSfromC

- 1 Installiert den mingw-w64 Compile unter c:Programme und öffnet mit den script mingw-w64.bat eine CMD Box, die den Pfad bereits um den 64-Bit-Compiler erweitert hat

Testet ob der richtige Compiler gestartet wird:
gcc --version

```
G:\mingw-w64>gcc --version
gcc (x86_64-posix-seh-rev0, Built by MinGW-W64 project) 8.1.0
Copyright (C) 2018 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

Figure 1: gcc-version

- 2 Übersetzt das Programm mit dem gcc Compiler.

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

- 3 Startet das erzeugte Programm in der CMD Box.

Selektiert das Programm AAB03-Name.exe
CPU:

Prozess	PID	Beschreib...	Status	Threads	CPU	Durchsc...	Betriebssyste...	Benutzername	Zyklus	Durchschnittl...	Plattform	Heraufgestuft
AAB03-Fabio.exe	13060	AAB03-...	Wird au...	1	0	0.00	Windows Vista	autoa	0	0.00	64 Bit	Nein

Figure 2: Programm-Ressourcenmonitor

- Welche PID hat das Programm?
13060
- Wie viele Threads laufen, wie viel CPU benötigt das Programm?
1

- Welche Module (Dateien und Bibliotheken) werden vom Programm verwendet?

Zugeordnete Handles			
Gefiltert von "AAB03-Fabio.exe"			
Prozess	PID	Typ	Handlename
AAB03-Fabio.exe	13060	Directory	\KnownDlls
AAB03-Fabio.exe	13060	File	\Device\ConDrv
AAB03-Fabio.exe	13060	File	\Device\ConDrv
AAB03-Fabio.exe	13060	File	\Device\ConDrv
AAB03-Fabio.exe	13060	File	\Device\ConDrv
AAB03-Fabio.exe	13060	File	C:\Users\autoa
AAB03-Fabio.exe	13060	File	\Device\ConDrv
AAB03-Fabio.exe	13060	Key	\REGISTRY\MACHINE\SYSTEM\Cont...
AAB03-Fabio.exe	13060	Key	\REGISTRY\MACHINE\SYSTEM\Cont...
AAB03-Fabio.exe	13060	Section	\Sessions\1\BaseNamedObjects\NLS...
AAB03-Fabio.exe	13060	Section	\Sessions\1\BaseNamedObjects\NLS...

Figure 3: Programm-Handles

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

Memory:

- Wie viel Speicher wird zugesichert/verwendet?
Es wird 480kB Zugesichert und 120kB verwendet.

Prozess	PID	Harte Fe...	Zugesichert (KB)	Arbeitssatz (KB)	Freigabe möglich (KB)	Privat (KB)
AAB03-Fabio.exe	13060	0	480	2.196	2.076	120

Figure 4: Programm-RAM

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

4.1 Wie geht das?

Im Ausgeführten Programm "c 1000" eingeben dies Reserviert dann $1000 * 1 \text{ MB}$ Blöcke im Arbeitsspeicher.

4.2 Von welcher Adresse bis zu welcher Adresse liegt der Speicherbereich?

```
1000 1M Blöcke reserviert ab Adresse: 00000000008BF040
bis Adresse: 00000003F0BF040
```

Figure 5: Programm-RAM-Reservierung

4.3 Kontrolliere die Größe mit einem Hex-Rechner

$000000003F0BF040 \rightarrow 1057747008$
 $000000000008BF040 \rightarrow 9171008$
 $1057747008 - 9171008 = 1048576000$



Figure 6: Byte in GB

4.4 Dokumentiert das Verhalten im Ressourcenmonitor.

Prozess	PID	Harte Fe...	Zugesichert (KB)	Arbeitssatz (KB)	Freigabe möglich (KB)	Privat (KB)
AAB03-Fabio.exe	13060	0	1.026.488	1.228	1.176	52

Figure 7: RAM-Reservierung-RM

4.4.1 Wird der Speicher bereits verwendet?

Nein der Speicher wird nicht verwendet.

4.4.2 Wieviel Speicher ist zugesichert, wie viel in Verwendung?

Ca. 1 GB ist zugesichert und wes werden ca. 1 MB verwendet

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

5.1 Was passiert im Ressourcenmonitor (Arbeitsspeicher)?

Prozess	PID	Harte Fe...	Zugesichert (KB)	Arbeitssatz (KB)	Freigabe möglich (KB)	Privat (KB)
AAB03-Fabio.exe	13060	0	1.026.488	638.096	1.176	636.920

Figure 8: W-1000

Das Programm verwendet jetzt ca. 600MB Memory.

5.2 Füllt zufällig weitere Speicherbereiche.

5.2.1 Was passiert?

Prozess	PID	Harte Fe...	Zugesichert (KB)	Arbeitssatz (KB)	Freigabe möglich (KB)	Privat (KB)
AAB03-Fabio.exe	13060	0	1.026.488	1.016.140	212	1.015.9...

Figure 9: W-4*1000

Es wurde noch 4 mal w 1000 ausgeführt. Es wird einfach der verwendete Speicher und den zugewiesenen Speicher angenähert. Bei jeder Ausführung wird weniger Memory verwendet.

6 Alloziert und beschreibt nun mehr als 4GB Speicher.

Prozess	PID	Harte Fe...	Zugesichert (KB)	Arbeitssatz (KB)	Freigabe möglich (KB)	Privat (KB)
AAB03-Fabio.exe	13060	0	4.617.516	360	236	124

Figure 10: 4.5GB-Rservierung

Es wird durch den Befehl c 4500 ca. 4.5GB reserviert

Prozess	PID	Harte Fe...	Zugesichert (KB)	Arbeitssatz (KB)	Freigabe möglich (KB)	Privat (KB)
AAB03-Fabio.exe	13060	0	4.617.516	512.360	236	512.124

Figure 11: 4.5GB-Verwendung

Nach der Durchführung des Befehls w 4500 wird nur 0.5GB verwendet. Es kann sein dass das Programm einfach keine freien RAM Plätze mehr findet da mein PC nur 4GB RAM besitzt.

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

Nein ein 32-Bit Anwendung kann maximal 4GB verwenden und nicht mehr.

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

7.1 Was erkennt man im Ressourcenmonitor?

7.2 Wie unterscheiden sich die beiden Prozesse?

Prozess	PID	Harte Fe...	Zugesichert (KB)	Arbeitssatz (KB)	Freigabe möglich (KB)	Privat (KB)
AAB03-Fabio.exe	13060	0	4.617.516	512.344	240	512.104
AAB03-Fabio.exe	10664	0	568	2.716	2.396	320

Figure 12: 2-Instanzen

8 Belege auch in der zweiten Programminstanz Speicherplatz.

8.1 Belege (allocate) dort mehrmals 1GByte und beschreibe den Speicher

Prozess	PID	Harte Fe...	Zugesichert (KB)	Arbeitssatz (KB)	Freigabe möglich (KB)	Privat (KB)
AAB03-Fabio.exe	10664	0	1.539.492	333.184	1.744	331.440
AAB03-Fabio.exe	9316	0	4.104.564	2.724	2.408	316

Figure 13: 2-Instanzen-RAM-Verwendung

8.2 Welcher Speicherbereich wird nun verwendet (Anfangs- und Endadressen)?

4000 1M Blöcke reserviert ab Adresse: 000000007FFE040
bis Adresse: 0000000179FFE040

Figure 14: 2-Instanzen-Speicherraum-1

Von: 7FFE040 → 2147475520
Bis: 179FFE040 → 6341779520
Ergibt: 4194304000Byte → 4,19GB

1000 1M Blöcke reserviert ab Adresse: 000000001FC65040
bis Adresse: 000000005E465040

Figure 15: 2-Instanzen-Speicherraum-2

Von: 1FC65040 → 533090368
Bis: 5E465040 → 1581666368
Ergibt: 1.048.576.000Byte → 1,04GB

8.3 Überschneidet sich der Speicherbereich tatsächlich mit dem anderen Prozess?

Keine Ahnung

8.4 Gib eine Erklärung dazu!

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

ja $4\text{GB} = 4 * 10^9$ Bytes

8.6 Wie wird das dem physisch im Rechner eingebauten Memory zugeordnet?

Durch die MMU (Memory Management Unit) ist dafür zuständig dass der logische Adressraum ohne Überschreitungen in den Physischen übernommen wird.

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

```
enter command:c 12000
12000 1M Blöcke reserviert ab Adresse: 000000007FFF6040
                                bis Adresse: 000000036DFF6040

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
```

Figure 16: Maximal-reservierbarer-Speicher

Es können maximal 12GB reserviert werden.

9 Was passiert wenn der physisch vorhandenen Speicher auf dem Rechner von den Prozessen aufgebraucht wird?

Es kann kein logischer Speicher mehr übernommen werden. Somit schreiben die meisten Programme die diesen physischen Speicher wirklich brauchen eine Fehlermeldung, dass das System zu wenig Speicher besitzt.

9.1 Wie sieht der Übersichtsbalken im Ressourcenmonitor aus? (screenshot)

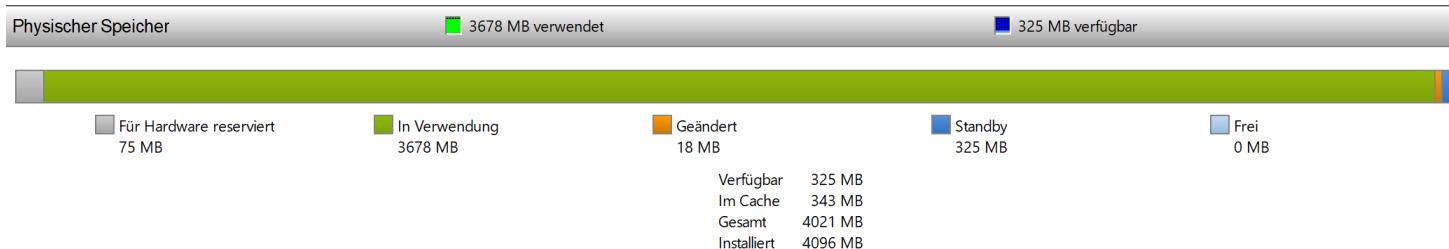


Figure 17: Arbeitsspeicher-Übersichtsbalken

Da mein Surface nur 4GB physischen RAM besitzt und ich versuche 12GB zu beschreiben werden nur 2 GB beschrieben und der Physische RAM somit steigt der grüne Balken "in Verwendung" ich habe aber auch beobachtet das anderen Programmen RAM entzogen wurde.

9.2 Was macht die Festplatte? Wie verhält sich der Rechner?

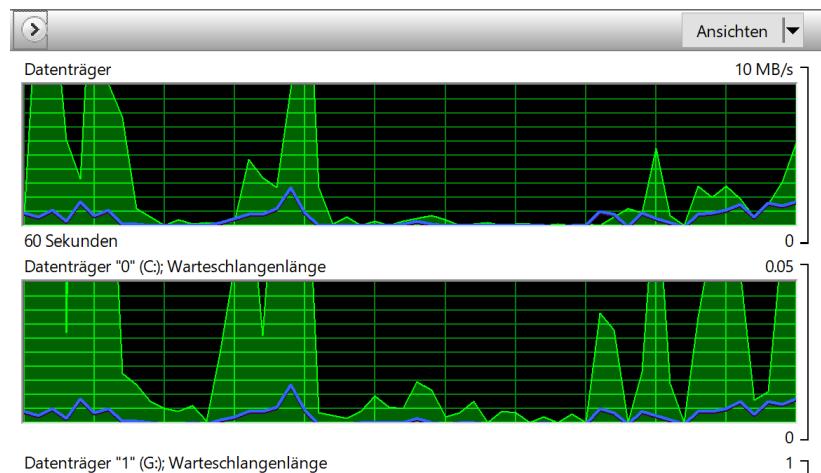


Figure 18: Datenträger-bei-beschreibung

Während der Beschreibung der Daten war die Festplatte kurz ausgelastet.

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

Der verwendete Arbeitsspeicher ändert sich kaum.

10 Was passiert, wenn Du eine der beiden Programminstanzen beendest?

10.1 Wie sieht die Speicherbelegung im Ressourcenmonitor dann aus? (Screenshot)

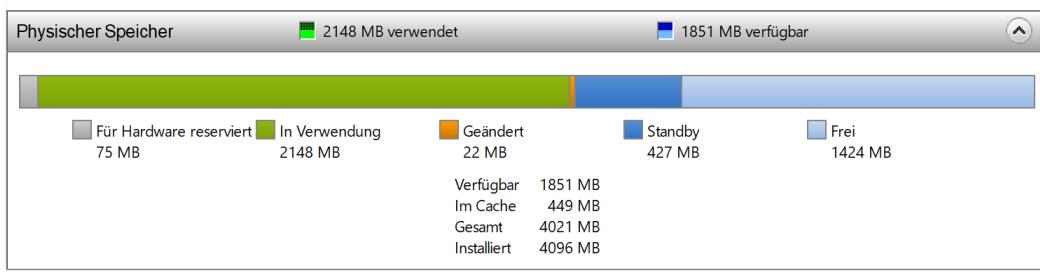


Figure 19: Speicher-nach-Prozess-Beendung

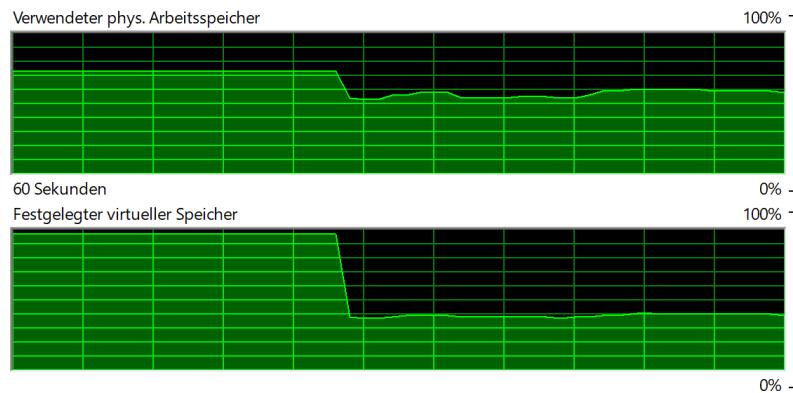


Figure 20: Speicher-nach-Prozess-Beendung

Der vorher belegte Speicher wird nicht mehr belegt. Somit wird auch weniger RAM verwendet.

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

11.1 Was beobachtest Du im Ressourcenmonitor?(screenshot)

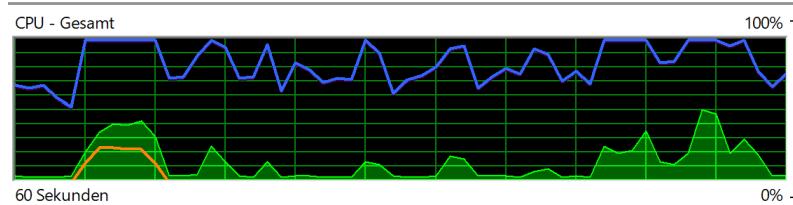


Figure 21: l-100-1

Es hat sich nur die Gesamtauslastung der CPU verändert, vor allem versucht die CPU durchgehend ihre maximal Frequenz zu erreichen obwohl die Auslastung dafür zu gering ist. Also könnte die CPU eigentlich um Strom zu sparen die Frequenz verringern.

12 Verwende die Funktion t (für mehrere Threads starten) mit 4 parallelen Threads:

12.1 Was beobachtest Du?

für eine Kurze Zeit wird die CPU zu 100% ausgelastet und der Prozess verwendet 5 Threads.

12.2 Wie sieht das im Ressourcenmonitor aus?

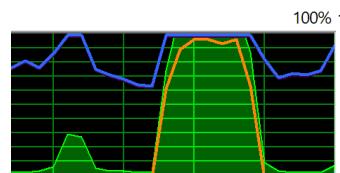


Figure 22: 4-Threads-Auslastung

Prozesse	115% CPU-Auslastung			115% Maximale Frequenz		
<input checked="" type="checkbox"/> Prozess	PID	Beschre...	Status	Threads	CPU	Durchsc...
<input checked="" type="checkbox"/> AAB03-Fabio.exe	10288	AAB03-...	Wird au...	5	87	14.96

Figure 23: 4-Threads-Prozess

12.3 Dokumentiere auch den Verlauf der CPU-Nutzung (Screenshot)

siehe 12.2

13 Was passiert, wenn Du 8 Threads oder mehr startest?

13.1 Wie verhält sich der Rechner?

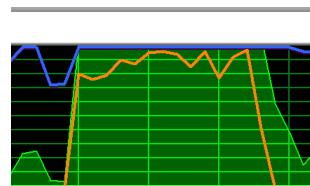


Figure 24: 8-Threads-Auslastung

Prozesse		115% CPU-Auslastung			115% Maximale Frequenz		
<input checked="" type="checkbox"/> Prozess	PID	Beschrei...	Status	Threads	CPU	Durchsc...	
<input checked="" type="checkbox"/> AAB03-Fabio.exe	10288	AAB03-...	Wird au...	9	95	17.87	

Figure 25: 8-Threads-Prozess

Der Rechner wird nicht langsamer und andere Programme scheinen kaum davon beeinflusst zu sein. Nur z.b Video schauen auf Youtube hakt ein bisschen.

13.2 Laufen die Threads bei 4 Hyperthreading Kernen tatsächlich parallel?

Ja. Hyperthreading sorgt dafür dass sich Threads Ressourcen teilen, die normalerweise ein CPU Kern berechnen müsste. Somit kann ein Thread die Ressourcen verwenden die ein anderer momentan nicht benötigt.

13.3 Wie wirkt sich das Starten von mehreren Threads auf die Laufzeit eines Threads aus? (Screenshot der CPU-Nutzung)

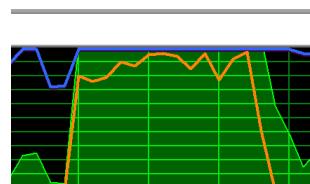


Figure 26: 8-Threads-Auslastung

Die Laufzeit ist länger.

14 Übersetzt das Programm auch in der virtuellen Linux-Mate-Maschine.

Vorsicht: Hier lautet der gcc Aufruf weil folgt: (Libraries nach dem C-File anfügen) gcc -o AAB03-walter AAB03-OSfromC.c -lpthread -lm

```
fabio@fabio-virtual-machine:~/Desktop$ gcc -o AAB03-Fabio osc.c -lpthread -lm
fabio@fabio-virtual-machine:~/Desktop$ ./AAB03-Fabio

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:■
```

Figure 27: Linux-Programm-Übersetzung

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

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

Neine da ich 1GB RAM der VM zugewiesen habe, kann ich maximal nur 700MB reservieren.

```
enter command:c 700
700 1M Blöcke reserviert ab Adresse: 0x7f537f824010
                                bis Adresse: 0x7f53ab424010

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 800
800 1M Blöcke reserviert ab Adresse: (nil)
                                bis Adresse: 0x32000000
```

Figure 28: Linux-maximal-reservierbarer-Speicher

15.2 Was macht die Ressource Memory und Swap (unter Resources)

Memory and Swap History

Figure 29: Linux-Memory-Swap

Nach dem Befehl w 1000 wird versucht den gesamten Speicher zu beschreiben dadurch ist der RAM zu 100% ausgelastet.

15.3 Was ist und wo liegt der SWAP?

SWAP ist der Auslagerungsspeicher der nicht nur genutzt wird wenn der RAM knapp wird. SWAP liegt dadurch auf der Festplatte. Es wird auch innerhalb großer Programme vom Kernel verwendet um Daten die höchstwahrscheinlich nicht gebraucht werden, aber trotzdem vom Programm in den RAM geschrieben werden, auszulagern. Man bekommt davon nichts mit da es Daten sind die nicht benötigt werden.