

Name, Matrikelnummer

Campus

Esslingen Flandernstraße

<b>Prüfer:</b>	Prof. Dr.-Ing. Rainer Keller	<b>Anzahl der Seiten:</b>	10
<b>Studiengänge:</b>	Softwaretechnik und Medieninformatik Technische Informatik	<b>Semester:</b>	SWB2 TIB2
<b>Klausur:</b>	<b>Betriebssysteme</b>	<b>Prüfungsnummern:</b>	IT 105 2004 2 SWB 3072 2 TIB 3072
<b>Hilfsmittel:</b>	keine, außer 1 DIN A4 Blatt, beidseitig von Hand selbst beschrieben	<b>Dauer der Klausur:</b>	90 Minuten

## Aufgabe 1: Allgemeines

**(9 Punkte)**

a) Wie waren die ersten elektrischen Computer aufgebaut?

 Mit Vakuumröhren, Relays  
 Verdrahtung

Trennung Berechnungseinheit und Speicher

4

b) Was kennzeichnet den Übergang zu modernen Rechner der 2. Generation?

 Prozessoren auf Halbleitertechnik auf Germanium oder Silizium  
 Transistoren

3

c) Welche Linux-Distribution haben wir in der Virtuellem Maschine genutzt?

Debian

2

## Aufgabe 2: Bash Shell

(14 Punkte)

a) Was machen die folgenden Bash Befehle?

<code>mkdir help</code>	den Ordner help erstellen
<code>strace ./program</code>	die System Calls von program verfolgen und ausgeben
<code>ps</code>	Zeigt Informationen zu allen Prozessen die kontrollierende Terminals haben
<code>rm -fr verzeichnis/</code>	"Verzeichnis" und alle Unterordner und Dateien ohne Nachfrage bei Problemen löschen
<code>wc datei</code>	Ausgabe der Wörter, Zeilen und Zeichen in "datei"
<code>bg</code>	sendet den aktuellen job in den Hintergrund
<code>grep datei text.txt</code>	sucht in "text.txt" nach "datei" und gibt alle Zeilen damit aus
<code>mount</code>	kann Dateisysteme einhängen, gibt alle eingehängten FS und Geräte aus
<code>mknod c 1 1 nod</code>	neuer Dateisystem node

9

b) Welche Betriebssystem-Tools müssen Sie hier verwenden?

Alle offenen Netzwerkverbindungen zeigen:	netstat
Module Informationen anzeigen:	modinfo
Module laden:	insmod oder modprobe
Einen Prozess „netter“ machen:	nice
Header in einer Binärdatei (Module) zeigen:	objdump -h

5

## Aufgabe 3: Hardware

(10 Punkte)

- a) Wie nennt man die Software eines Betriebssystems, welche bestimmte Hardware, bspw. einen USB-Stick ansprechen?

Ein Kernelmodul, ein (Hardware-)Treiber, diese können als Module eingebunden oder fest im Kernel integriert sein

2

- b) Nennen Sie Ihnen bekannte systemnahe Programme (mindestens zwei) um Hardware des Rechners herauszufinden?

lsusb, lspci, cat /proc/cpuinfo

2

- c) Meine Hardware tut nicht, wo finde ich mehr Informationen raus?

dmesg, journalctl -xe, lsmod

4

- d) Was gehört zum Betriebssystem, was nicht?

dazu: alles das notwendig ist die Hardware zu abstrahieren, Kernel, Module, Treiber  
nicht dazu: alles was der Nutzer darauf ausführen möchte, systemnahe Programme, Editoren etc

2

## Aufgabe 4: Systemaufrufe (19 Punkte)

- a) Welche Möglichkeiten gibt es auf x86-Prozessoren (32-Bit und 64-Bit), Funktionen im Linux-Betriebssystem aufzurufen?

syscall, sysenter, glibc, SW-Interrupt x80

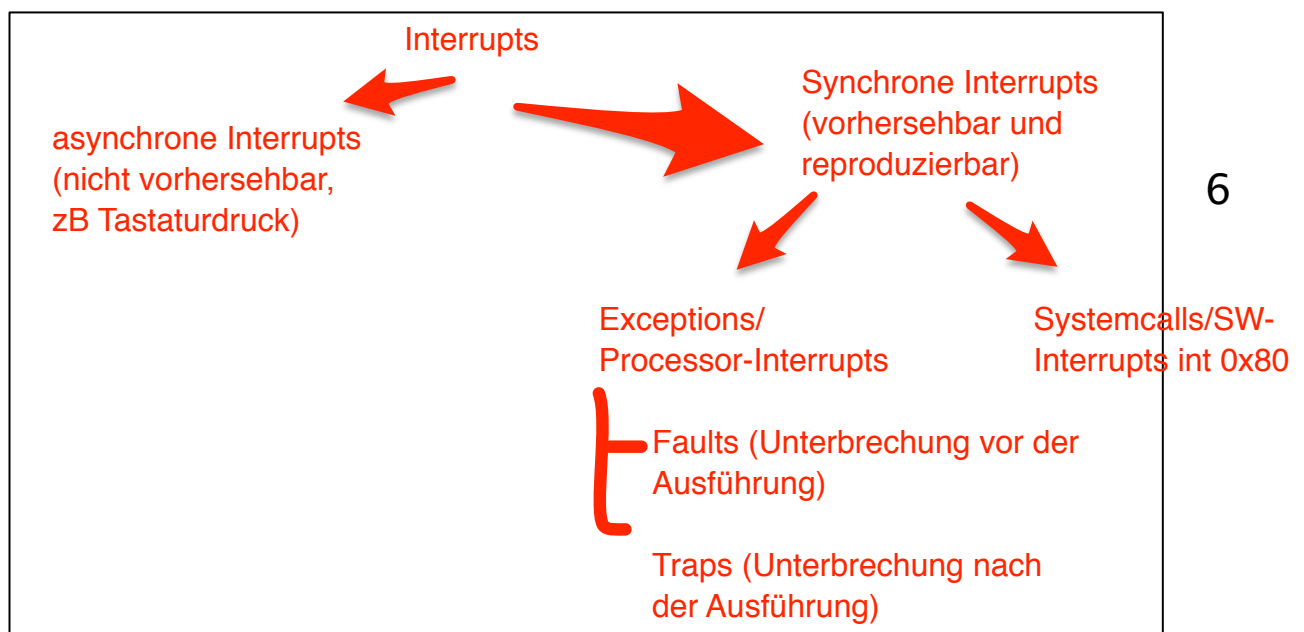
3

- b) Beschreiben Sie den Ablauf eines Hardwareinterrupts anhand eines Tastendrucks auf dem Keyboard ihres PCs?

1) Tastendruck -> Tastaturgerät meldet Interrupt auf Bus (von Interrupt Controller, CPU und Geräten) -> Interrupt Controller entscheidet was getan werden muss, wenn gerade ein anderer Interrupt bearbeitet wird oder auf einem höherpriorisierten Bus ein Interrupt liegt, dann bleibt das Signal solange auf dem Bus, bis es durch den Prozessor behandelt wird. -  
> IC erzeugt Interrupt auf Bus  
2) Mikroprozessor wird unterbrochen  
3) aktueller Prozess merkt davon nichts  
4) ISR wird geladen und liest die Daten vom Gerät in Puffer  
5) Betriebssystem springt in den unterbrochenen Prozess zurück  
6) Periodisch muss der OS-interne Puffer kopiert werden  
7) Wartet ein Prozess auf den Tastendruck im Puffer kann er jetzt wieder ausgeführt werden

6

c) Zeichnen Sie die Interrupt-Klassifikation auf:



6

d) Wie lange dauert ein Systemaufruf circa? Und wieso war getpid() so schnell?

mehrere Hundert Taktzyklen

getpid() war so schnell, weil die pid in einem Cache lag und von dort kopiert werden konnte

4

## Aufgabe 5: Virtueller Speicher (18 Punkte)

- a) Welche beiden Eigenschaften müssen für Speicherzugriffe gelten, damit Caches optimal funktionieren? (bitte erklären)

die relevanten Daten müssen im Cache liegen,  
nur wenn das, was am wahrscheinlichsten  
Gebraucht wird im Cache liegt, wird die  
erwartete Speicherzugriffzeit kürzer  
temporal, in kurzer Zeit wird wahrscheinlich wieder auf den selben oder in  
der Nähe wieder auf Speicher zugegriffen

4

partial, mit hoher Wahrscheinlichkeit die wieder in der Nähe zugreifen

- b) Wie viele Bits bietet der Intel Prozessor für Schutzebenen, wie viele Ebenen erlaubt dies und wie viele nutzt Linux?

1.	Wie viele Bits?	3
2.	Wie viele Ebenen?	4
3.	Linux nutzt?	2

3

- c) Welche Speicherseitengrößen unterstützen 64-Bit Intel & AMD CPUs?

4KiB, 2/4MiB, 1GiB

3

- d) Der Buddy-Allokator erlaubt, sehr schnell freie Speicherbereiche zu identifizieren. Die untenstehende Ansicht entspricht der Darstellung von Wikipedia. Zuerst ist der Speicher komplett frei. Zeichnen Sie die folgenden Allokationen ein:
- 1. Programm A alloziert 17 kB Speicher
  - 2. Programm B alloziert 3 kB Speicher
  - 3. Programm A alloziert 13 kB Speicher

	4kB	4kB	4kB	4kB	4kB	4kB	4kB	4kB	4kB	4kB	4kB	4kB	4kB	4kB	4kB	4kB
1.	2 <sup>4</sup>															
2.																
3.																
4.																
5.																
6.																
7.																
8.																
9.																
10.																
11.																

## Aufgabe 6: Linux Kernel (13 Punkte)

a) Wohin werden Linux Kernel Module Dateien installiert?

`/lib/modules/VERSION/`

2

b) Erklären Sie die Zeilen der Ausgabe von `lsmod`:

usb_storage	39350	1	
aes_intel	17938	0	
aes_i586	16956	1	aes_intel

Andere Module die dieses Modul benötigen

8

name des moduls

Größe in Bytes

wie häufig es von anderen Modulen oder Prozesse es gerade verwendet wird



---

Name, Matrikelnummer

- c) Circa wie groß ist der Linux Kernel in Lines-of-Code und in welcher Programmiersprache ist der geschrieben?

C, 20Mio LoC

3

## Aufgabe 7: IPC

(9 Punkte)

- a) Welches ist die schnellste Art der Interprozesskommunikation zwischen Prozessen eines Rechners und warum?

Sich Gegenseitig in den RAM schreiben, Memoymaps / Sharedmemory  
Argumentation?

process\_vm\_read und process\_vm\_write bei Elternkindprozessen

5

- b) Warum sind Dateien keine gute Form der Interprozesskommunikation?

Langsam, man muss auf Platte schreiben, was wenn ein zweiter Prozess  
auch auf die Datei zugreift

4

## Aufgabe 8: Dateisysteme

(8 Punkte)

- a) Welche Dateisysteme haben wir in der Vorlesung behandelt?

NTFS, ExtFS, FAT

2

- b) Was zeichnet das Dateisystem vom alten MS-Dos aus und wieso wird es immer noch verwendet?

es kann keine sehr großen Dateien beinhalten  
es ist ein sehr einfaches Dateisystem und deshalb können auch  
embedded Devices wie zB mp3 Sticks gut damit umgehen

6