

Name, Matrikelnummer \_\_\_\_\_

<b>Prüfer:</b>	Prof. Dr.-Ing. Rainer Keller	<b>Anzahl der Seiten:</b>	9
<b>Studiengänge:</b>	Softwaretechnik und Medieninformatik Technische Informatik Ingenieurpädagogik	<b>Semester:</b>	SWB2 TIB2 IEP2
<b>Klausur:</b>	<b>Betriebssysteme</b>	<b>Prüfungsnummern:</b>	IT 105 2004
<b>Hilfsmittel:</b>	keine, außer 1 DIN A4 Blatt, beidseitig von Hand selbst beschrieben	<b>Dauer der Klausur:</b>	90 Minuten

Bitte lesen Sie die Aufgaben sorgfältig durch. Jede Aufgabe besteht aus Unteraufgaben – für die es Teilpunkte gibt. Jeder Punkt entspricht ca. 1 Minute Arbeitszeit. Nutzen Sie also den zur Verfügung stehenden Raum und die Zeit aus, um möglichst sorgfältig und ausführlich zu antworten.

## Allgemeines

**(9 Punkte)**

a) Was war so exorbitant an der Entwicklung der Mikroprozessoren?

- Moore's Law: - Anzahl d. Transistoren verdoppelt sich alle 12 Mon.

3

b) Wie lautet die in der Vorlesung gegebenen Definition eines Betriebssystems?

4

c) Circa wie groß ist ein L3 Cache eines modernen Intel Prozessors?

Ca. 20-50 MB an L3 Cache

2

Name, Matrikelnummer

**Bash Shell****(17 Punkte)**

a) Benennen Sie für die jeweilige Shell-Funktion den passenden Shell-Befehl:

Größe aller Dateien im Verzeichnis:	du -h
Hilfeseite von bash aufrufen:	man bash
Prozess 666 hart beenden:	Kill -9 666
Top Prozesse auflisten:	top oder htop, powertop
Datum ausgeben:	date
Archive ver-/auspacken:	tar
Prozess in den Hintergrund:	bg
Zeilen zählen:	wc -L

4

b) Die Umgebungsvariable DATEI enthält den Namen einer Datei. Aus dieser Datei sollen Sie Vorname, Name, Matrikelnummer und Note von Studenten auslesen. Die Daten sind jeweils durch Doppelpunkte getrennt, die Note ist zweistellig, bspw.

Max:Meier:12345:20

Moritz:Mueller:23456:1

Miriam:Musterfrau:34567:10

Schreiben Sie ein Shell-Skript, welches die Matrikelnummer des Studenten mit der besten Note ausgibt.

```
#!/bin/bash
BESTE=99

while read LINE; do
    MATR='echo $LINE | cut -f3 -d ':' ;
    NOTE='echo $LINE | cut -f4 -d ':' ;
    echo "Matrikelnummer: $MATR Note: $NOTE";
    if test $ NOTE -le $BESTE; then
        BESTE=$NOTE
        BESTE_M=$ MATR
    fi
done < $DATEI
echo " Beste Note $BESTE von Student mit $ BESTE-M "
```

Prüfen ob DATEI nullgesetzt ist  
if [x \$DATEI x = "xx"]; then

7

\_\_\_\_\_  
Name, Matrikelnummer

c) Was sind systemnahe Programme? Bitte geben Sie Beispiele an.

Programme, die eng mit dem System zusammenarbeiten

3

d) Was machen die beiden folgenden Programme?

mount	
fsck	
mknod	erzeugt spezielle Geräte-datei in /dev

3

---

Name, Matrikelnummer**Scheduling und Systemcalls****(22 Punkte)**

- a) Nennen Sie drei Optionen zum Setzen der Häufigkeit der Unterbrechung der CPU durch den Linux Scheduler – und was sind deren Vor- / Nachteile

Wie häufig?	Vor- / Nachteil
1.	
2.	
3. variable	- niedriger Stromverbrauch

5

- b) Welche beiden (temporären) Informationen verarbeitet ein CPU-Scheduler, um die Priorität eines Tasks neu zu berechnen? Wieso priorisiert er bestimmte Tasks höher und welche Tasks sind das?

- Wie rechenintensiv  
- Wie I/O-intensiv  
I/O-intensiv werden höher prior gewertet

3

- c) Wie kann man Systemcalls einer Anwendung mit der PID 123 mitlesen?

strace -p 123

2

Name, Matrikelnummer \_\_\_\_\_

d) Beurteilen Sie jede Aussage ob sie Wahr (W) oder Falsch (F) ist:

Aussage	W/F?
1:1 Modell heißt ein User-Thread ist ein Linux-Task	W
Das 1:1 Modell reduziert Komplexität	W
Früher hatte Linux ein 1:n Task:Thread Modell	F
Zur besseren Lastbalance werden Tasks auf freie Cores verschoben	W
Die <code>vruntime</code> ist in Millisekunden (ms) aufgeteilt (in ns)	F
In <code>struct task{}</code> stehen die lauffähigen und gestoppten Tasks Informationen zu einem Task	F
Ein Task läuft nachdem er auf einem core gestartet wurde nur auf diesem	F
Ein Prozess besteht aus einem Thread und ist ein Task	F
Die CPU-Zeit wird an die <code>nr_running</code> Tasks verteilt	W

5

e) Welche Ressourcen teilen sich alle Threads eines Prozesses, welche sind pro Thread?

<p>Geteilte Ressourcen?</p> <ul style="list-style-type: none"> <li>- TCP- Sockets</li> <li>- Speicher / Heap</li> <li>- Signal Handler</li> </ul>	
<p>Per-Thread Ressourcen?</p> <ul style="list-style-type: none"> <li>- Stack</li> <li>- Register Set</li> <li>- Priority</li> </ul>	

3

f) Malen Sie die Zuordnung von Linux Threads eines Prozesses auf Tasks, sodaß die Zugehörigkeit zu den Ebenen klar wird:

4

Name, Matrikelnummer

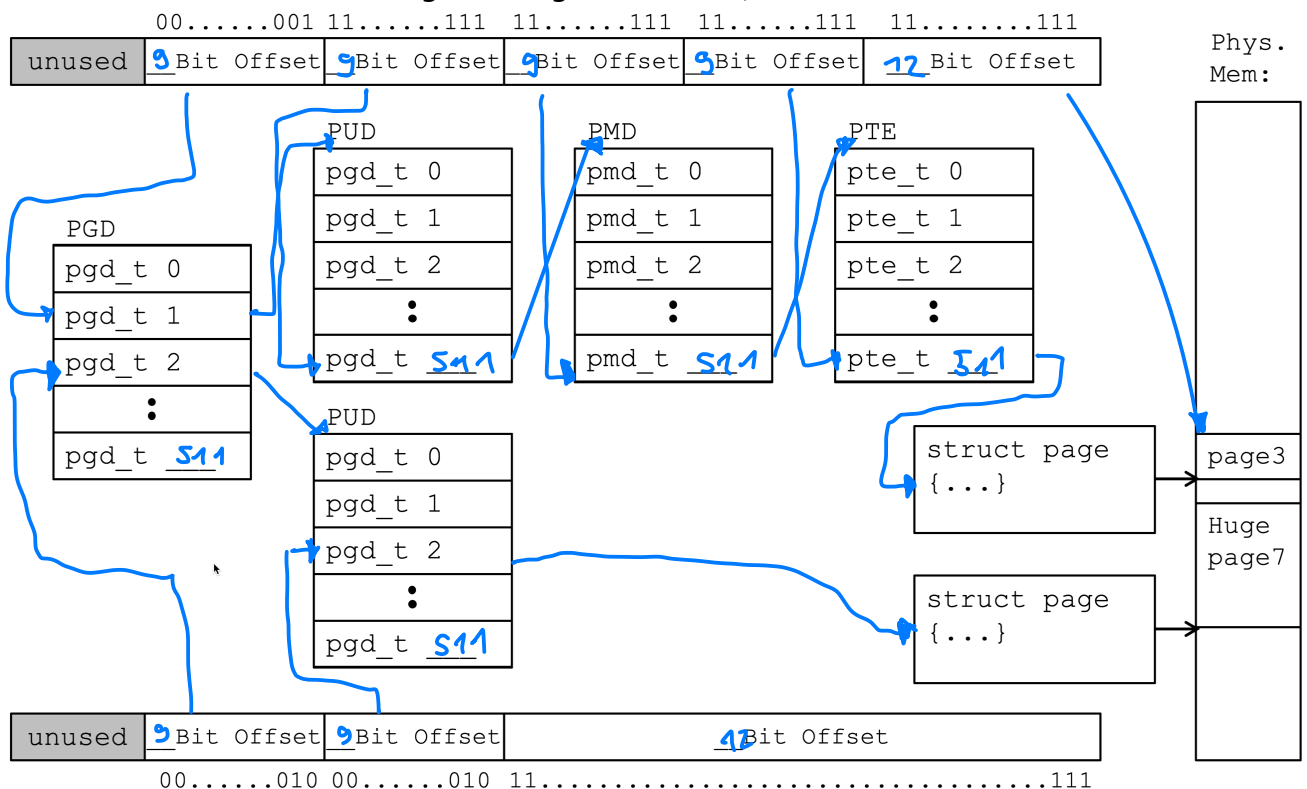
## Virtueller Speicher

(19 Punkte)

- a) Zeichnen Sie die Indexe und Pointer ein für die folgenden 64-Bit Virtuellen Adressen – bitte **beachten** Sie die binären Zahlenwerte (0...001 bedeutet eine 1 im niederwertigsten Bit, ansonsten Nullen). Geben Sie weiterhin die Anzahl Bits an den **unterstrichenen** Stellen ein.

8

Sollten Sie Offset-Längen korrigieren wollen, nutzen Sie bitte das Feld unten.



Korrektur:

Name, Matrikelnummer

- b) Der Data Segment Descriptor bietet 3 Bits für Speicherschutz: Writable, Accessed und Present. Welche Konzepte lassen sich mit diesen Bits umsetzen?

Speicherschutz  
Copy on Write

Access bit: wird es aktiv verwendet? ist die Page wichtig  
Present bit: Ist es gerade im RAM? Oder muss von Festplatte eingelesen werden

4

- c) Der Intel Prozessor unterstützt 4 Ebenen für den Privilege Level (CPL). Welche Softwareschicht auf einem Betriebssystem beispielhaft zugeordnet sein?

0. Ebene:	Kernel (Microkernel) Hypervisor
1. Ebene:	Treiber (HW-Treiber)
2. Ebene:	Systemnahe Programme, Dateisysteme, Netzwerkschnittstellen
3. Ebene:	User-Code

4

- d) Der Buddy Allocator für Speicherverwaltung hat ein Problem mit Fragmentierung. Wieso ist das für Verwaltung von VM-Pages kein Problem?

- Buddy Allocator kann freie Speichersystem wieder zusammenschieben

3

---

 Name, Matrikelnummer

## Interprozesskommunikation

## (14 Punkte)

a) Bitte geben Sie min. einen Unix-Funktionsaufruf je Kommunikationsmodell an:

Direkt in Speicher eines Prozesses schreiben:	<code>vmprocess readv</code>
Einem Prozess eine Benachrichtigung schicken:	<code>signal, kill</code>
Datei beschreiben:	<code>write(), open(), read(), close()</code>
Uni-direktionale Verbindung:	<code>pipe()</code>

2

b) Warum könnte eine Uni-direktionale Verbindung langsamer sein – aber wieso ist das Konzept dennoch so gut?

Gründe für langsame Performance: - im Kernel wird gepuffert SystemCall <code>write()</code> , <code>read()</code>
Gründe für gutes Konzept: - einfache Architektur

6

c) Nennen Sie den vermutlich schnellsten IPC-Kommunikation bezüglich Bandbreite und eine Begründung warum:

Schnellster Call:	Begründung:
<code>mmap()</code>	

6



Name, Matrikelnummer

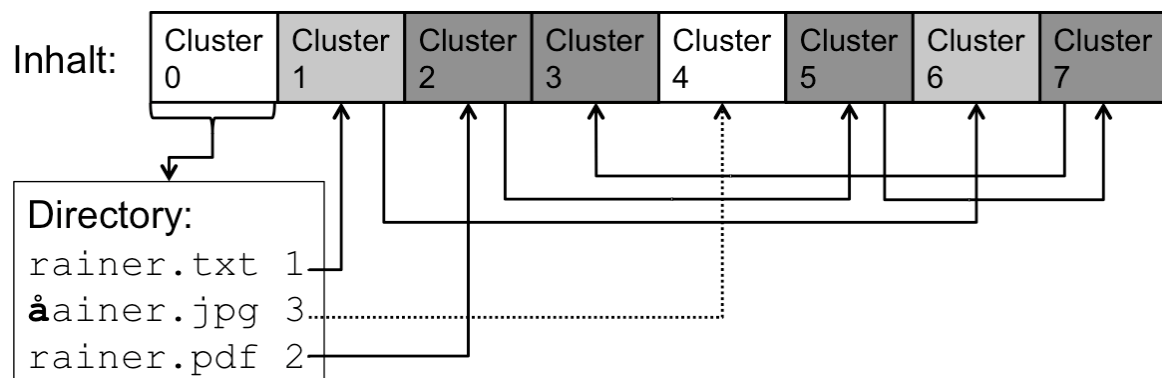
**Dateisysteme****(9 Punkte)**

- a) Zur korrekten Unterscheidung von Datenträgergrößen wurde eine IEC-Norm eingefügt. Nennen Sie mindestens 3 Größen und geben grob an was das an Unterschied ausmacht.

Bezeichnung	Prozentualer Unterschied

3

- b) Defragmentieren Sie das unten stehende FAT-Verzeichnis. Geben Sie Schritte mit an, sowie die Folge der Cluster je Datei.



1.	2.	3.
4.	5.	6.
Resultat für Folge rainer.txt		
Resultat für Folge rainer.pdf:		

6