

Name, Matrikelnummer

Prüfer:	Prof. Dr.-Ing. Rainer Keller	Anzahl der Seiten:	10
Studiengänge:	Softwaretechnik und Medieninformatik Technische Informatik Ingenieurpädagogik	Semester:	SWB2 TIB2 IEP2
Klausur:	Betriebssysteme	Prüfungsnummern:	IT 105 2004
Hilfsmittel:	keine, außer 1 DIN A4 Blatt, beidseitig von Hand selbst beschrieben	Dauer der Klausur:	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

(5 Punkte)

- a) Aus welchen Komponenten bestanden Rechner der 1. Generation (z.B. Zuse Z1)
- b) Circa wie groß ist der Linux Kernel in Lines-of-Code und in welcher Programmiersprache ist der geschrieben?
- c) Nennen Sie vier komplett unterschiedliche Betriebssysteme:

1. Betriebssystem:	3. Betriebssystem:
2. Betriebssystem:	4. Betriebssystem:

Name, Matrikelnummer

Bash Shell

(20 Punkte)

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

Systemcalls anzeigen:	
Netzwerk konfigurieren:	
Link erstellen:	
Dateisystem erstellen:	
Zeile ausgeben:	
Archive ver-/auspacken:	
Prozess in den Hintergrund:	
Signal an Prozess senden:	

b) Die Datei `gehaltsliste.txt` enthält die Namen, Stundenlohn und Arbeitsstunden durch Doppelpunkte getrennt (CSV) von Mitarbeitern einer großen Abteilung. Schreiben Sie ein Shell-Script, welches die Anzahl der Mitarbeiter mit 3-stelligem Lohn zählt, **sowie** den Mittelwert der Gehälter berechnet.

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

d) Was bedeuten die beiden Umgebungsvariablen `PATH` und `LD_LIBRARY_PATH`?

<code>PATH</code>	
<code>LD_LIBRARY_PATH</code>	

e) Schreiben Sie die passenden Umgebungsvariablen hin:

Vorheriger Pfad:	
Prompt String:	

Name, Matrikelnummer

Verwendete Shell	
Pfad der Doku:	
Erster Parameter einer Funktion	
PID der Bash	

Name, Matrikelnummer

Scheduling und Systemcalls

(19 Punkte)

- a) Wie lange dauert ein Systemaufruf circa und wieso war `getpid()` so schnell?

Wie lange?

Was war mit `getpid()`?

- 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?
- c) Was passiert, wenn ein Task sein Zeitkontingent aufgebraucht hat und was passiert, wenn kein Task mehr Zeitkontingent übrig hat?

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	
Das 1:1 Modell reduziert Komplexität ist aber langsam	
Der Linux Kernel implementiert den Desolate Fair Scheduler (DFS)	
Der Linux Scheduler läuft bei jedem Interrupt	
Zur besseren Lastbalance werden Tasks auf freie Cores verschoben	
Wenn ein Task blockiert, läuft der Scheduler	
Das Zeitkontingent für alle Threads eines Tasks ist <code>vruntime</code>	
Die <code>vruntime</code> ist in Millisekunden (ms) aufgeteilt	
In <code>struct rq{}</code> stehen die lauffähigen und gestoppten Tasks	
Die Migration eines Tasks von einem Core auf einen anderen ist teuer	
Der Linux Scheduler läuft effizient in $O(n)$	
Die CPU-Zeit wird an die <code>nr_running</code> Tasks verteilt	

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

Geteilte Ressourcen?
Per-Thread Ressourcen?

Name, Matrikelnummer

Virtueller Speicher

(17 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.

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

Name, Matrikelnummer

b) 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 (mit Unterschriften) 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.	2^4															
2.																
3.																
4.																
5.																
6.																
7.																
8.																
9.																
10.																
11.																

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

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

Name, Matrikelnummer

Linux Kernel Module

(10 Punkte)

- a) Programmieren Sie ein minimales Kernel-Modul, welches beim Laden prüft ob der ladende Task die FPU nutzt und dies ausdrückt. Die relevante Datenstruktur `task_struct` sowie die `PF_*` Flags (beide aus `include/linux/sched.h`) sind:

```
struct task_struct { ...
    volatile long state; // -1 unrunnable, 0 runnable, >0 stopped
    unsigned int flags; // Defined by PF_* below
... }
#define PF_IDLE          0x00000002
#define PF_USED_MATH     0x00002000 // If set the fpu was used
```

- b) Ihr Modul soll den Parameter „`alloc_size`“ als Integer beim Laden übernehmen können. Schreiben Sie hierzu den C Code:
- c) Bitte geben Sie den Befehl an, um ihr Modul `modfpu.ko` zu laden:

Name, Matrikelnummer

Interprozesskommunikation

(8 Punkte)

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

Uni-direktionaler Daten-transfer via Kernel	
Gemeinsamer Speicher	
Asynchrone Benachrichtigung eines Events	
Direkt in den Speicher eines Prozesses schreiben	

b) Nennen Sie Vor- und Nachteile Daten zwischen Prozessen mittels Dateien auszutauschen:

Vorteile:
Nachteile:

Name, Matrikelnummer

Virtualisierung, Echtzeit & andere OS (11 Punkte)

a) Nennen Sie die 4 Klassifikationen für Antwortverhalten:

1.	
2.	
3.	
4.	

b) Warum ist der Linux Kernel in der Standardkonfiguration nicht echtzeitfähig?

c) Worauf basiert das Betriebssystem MacOS und wie heißt der Kernel?

Basis	
Kernel:	

d) Wie unterscheidet sich Microsoft Windows Server von einem Standard Windows 10?