

Name, Matrikelnummer _____

Prüfer:	Prof. Dr. Tobias Heer	Anzahl der Seiten:	12
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 (keine Kopie). Das Hilfsmittel ist abzugeben. Sollte kein Hilfsmittel verwendet werden ist dies in der Klausur anzuzeigen.		
		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. Achten Sie auf Schlüsselwörter wie „in Stichworten“, „kurz“, „ausführlich“, „nennen“, „erklären“ oder „im Detail“. Der Umfang Ihrer Antwort soll sich danach richten.

1. Allgemeines

(13 Punkte)

- a) Erklären Sie weshalb ein einfaches eingebettetes System (z.B. ein Arduino) auch ohne ein dediziertes Betriebssystem (z.B. Linux/FreeRTOS) funktionieren kann und welche Eigenschaften das System ohne Betriebssystem nicht hat. Nennen Sie mindestens zwei Eigenschaften und erklären Sie kurz weshalb diese nicht gegeben sind und nicht benötigt werden.

Chat GPT

Für viele Anwendungen reicht es aus, die Hardware und Peripherie direkt anzusteuern, ohne ein dediziertes Betriebssystem, dass den Zugriff auf Speicher, Rechen- und Kommunikationsmittel sicher regelt, dabei gibt es aber Eigenschaften, die ohne BS in der Regel nicht zur Verfügung steht:

- Multitasking: einfach eingebettetes System kann keine parallelen Prozesse gleichzeitig ausführen → keine Multitasking-Umgebung
- Ressourcenmanagement: effiziente Verwaltung der Ressourcen, wie Priorisierung der Aufgaben und Zuweisung der Speicherbereiche ohne BS schwer möglich

Da eingebettete Systeme meistens nur bestimmte Aufgaben erfüllen müssen und wenig Ressourcen haben wird ein dediziertes Betriebssystem nicht benötigt.

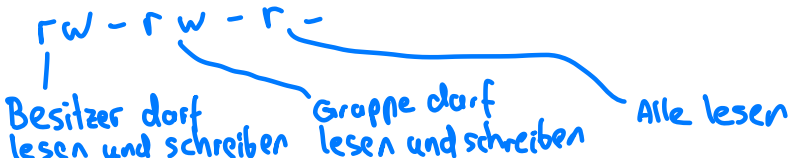
6

- b) Was findet sich im /proc Verzeichnis unter Linux. Inwiefern ähneln und unterscheiden sich die Einträge dieses Verzeichnis von denen anderer Verzeichnisse (z.B. /etc). Geben Sie in Ihrer Erklärung mindestens ein Beispiel für eine Information im /proc Verzeichnis.

/proc-Verzeichnis ist ein virtuelles Dateisystem, das Dateien mit aktuellen Informationen über das System und Prozesse enthält (wie z.B. Interrupts)
Andere Verzeichnisse wie z.B. /etc sind statisch, d.h. sie enthalten keine aktuellen Informationen. In diesem Fall enthält /etc verschiedene Konfigurationsdateien

3

- c) Ein Eintrag wird im Verzeichnis-Listing mit folgenden Informationen angezeigt:
-rw-rw-r-- 2 admin admin 67 Nov 29 12:15 admin
Erklären Sie alle Bestandteile dieser Darstellung und was diese für diesen konkreten Eintrag bedeuten.

-	File
rw-rw-r--	
2	Anzahl Verweise
admin	User
admin	Group
67	Größe Bytes
Nov 29 12:15	Datum der letzten Änderung
admin	Name der Datei

4

Name _____

2. Bash**(22 Punkte)**

- a) Erklären Sie für jedes der folgenden Symbole die Bedeutung in der Linux Toolchain bzw der Bash. Gemeint ist nicht das Literal. Das heißt „&“ steht also z.B. nicht für „das Zeichen welches ein Kaufmanns-Und darstellt“, sondern die besondere Funktion die & in der Kommandozeile hat.

	Die Standardausgabe eines Befehls in die Standardeingabe eines anderen Befehls weiterleiten	4
&	Programm im Hintergrund starten	
2>1	Fehlerausgabe eines Programms in Standardausgabe umleiten	
\$\$	Process ID des aufrufenden Skript	
*	Wildcard für beliebige Zeichenkette (null oder mehrere)	
	Verknüpfung von Kommandos (Bspw. Befehl 1 Befehl 2, Befehl 2 wird ausgeführt, Befehl 1 mit einem Fehler abgeschlossen wurde).	
>	Sonderzeichen zum umleiten Ein- / Ausgabe	
\$1	Script: 1. übergebener Parameter	

- b) Erklären Sie was Symbolic- und Hard-Links sind und wie sie sich unterscheiden. Gehen Sie auch darauf ein, was jeweils beim Löschen und Umbenennen des Ziels geschieht.

Hard-Links verweisen auf andere Dateien innerhalb des Dateisystems. Symbolic-Links verweisen auf andere Dateien / Verzeichnisse auch über Dateisystem-grenzen hinaus. Hard-Links erzeugen zusätzlichen Verzeichniseintrag, der auf i-node der ursprünglichen Datei zeigt, Daten sind aber einmal auf Festplatte gespeichert. Symbolic-Links hingegen erzeugen auch einen zusätzlichen Verzeichniseintrag, dieser zeigt aber auf eigenen i-node, welcher auf Pfad (bzw. Verzeichniseintrag) der ursprünglichen „gelinkten“ Datei zeigt.

Wird das Ziel gelöscht bleibt der Hard link unverändert bestehen, lediglich die Anzahl der Verweise wird um eins heraufgesetzt. Der Symbolic-Link zeigt ins Leere (Toter Link), er weiß nicht ob das Ziel noch existiert. Beim Umbenennen des Ziels bleibt der Hard link mit dem Ziel verlinkt, der Symbolic-Link zeigt wieder ins Leere => nicht zu erkennen ob Ziel verändert wurde oder ob Ziel nicht mehr existiert.

- c) In einem Verzeichnis liegen die Gehaltslisten der einzelnen Abteilungen eines Unternehmens. Die Gehaltslisten sind folgendermaßen strukturiert: „Name;Vorname;Gehalt;Betriebszugehörigkeit-in-Monaten“.

Zum Beispiel:

Mustermann;Max;3400;22

Beispielfrau;Melanie;4700;12

Es liegen mehrere dieser Dateien im Verzeichnis – für jede Abteilung gibt es eine Datei. Die Dateinamen sind „[Abteilung].csv“. Zum Beispiel:

Entwicklung.csv Buchhaltung.csv

Schreiben Sie ein Bash Script welches errechnet, welche die Abteilung mit den höchsten Lohnkosten ist und den Namen dieser Abteilung zurückgibt. Es sollen dabei beliebig viele Einträge pro Datei und beliebig viele csv Dateien möglich sein.

14

```
#!/bin / bash
max_kosten=0
Abteilung=""
for file in ./*.csv do
    kosten=$(awk -F ';' {sum+=$3} END {printf "%d",sum}' "$file")
    if ((kosten > max_kosten)); then
        highest_cost = $total_cost
        Abteilung = $(basename "$file".csv)
    fi
done
echo "Abteilung mit den höchsten Lohnkosten: $Abteilung"
```

Name

Platz für weiteren Code:

3. Scheduling und IPC

(14 Punkte)

- a) Beschreiben Sie detailliert, wie die Wahl der Zeitscheibe beim Scheduling das Verhalten der folgenden Systeme beeinflusst: a) Serversystem mit vielen parallelen Clients, b) Rechensystem mit wenigen parallelen Jobs. Gehen Sie auch darauf ein, in welcher Größenordnung die Zeitscheiben jeweils gewählt werden sollte (in ms) und warum?

- a) Beim Serversystem mit vielen parallelen Clients ist wichtig, dass jeder einzelne Client eine angemessene Zeitscheibe erhält. Die Zeitscheibe sollte nicht zu kurz sein, da dies zu häufigen Wechseln zwischen Clients führen kann, was wiederum unnötig Ressourcen verbraucht. Zeitscheibe sollte aber auch nicht zu lang sein, da es möglich wäre, dass ein einzelner Client zu lange die Ressourcen bezieht und die anderen Clients zu kurz kommen. Die Zeitscheibe sollte ungefähr 1-10 ms betragen, damit jeder Client ausreichend Rechenzeit bekommt.
- b) Beim Rechensystem mit wenigen parallelen Jobs steht der Fokus auf der Geschwindigkeit der Durchführung eines Jobs, d.h. die Wahl der Zeitscheibe sollte ausreichend sein, damit die Jobs erfolgreich durchgeführt werden können. Sprünge zwischen den Jobs sind wenig sinnvoll, da sie sonst unterbrochen und wiederhergestellt werden müssen. Je nach dem wie lange die Jobs brauchen, ist es sinnvoll mehrere ms bis zu einigen Sekunden als Zeitscheibe zu wählen.

5

- b) Weshalb macht es Sinn, dass ein Scheduler I/O-intensive Prozesse bevorzugt behandelt und was wäre die Folge, wenn er dies nicht tun würde. Erklären Sie.

Da I/O-intensive Prozesse lange Wartezeiten mit sich bringen (wegen Interaktion mit Peripherie/Benutzer). Diese Wartezeit kann genutzt werden, z.B. anderen Prozess mit vielen Rechenoperationen bekommt CPU in der Zeit. Die Folge wäre, dass Ressourcen verschwendet würden (CPU untätig) und sich das System verlangsamen würde.

5

Name _____

- c) Welche Ressourcen teilen sich alle Threads eines Prozesses, welche sind pro Thread vorhanden?

Geteilte Ressourcen?

- Offene File-Deskriptoren (FDs) (Dateien, TCP-Sockets, etc)
- Speicher (alles was alloziert wurde)
- Signal-Handler

Per-Thread Ressourcen?

- Registersatz
- Stack
- Thread-ID, Priority
- errno

4

4. Virtueller Speicher

(20 Punkte)

- a) Erklären Sie das Konzept von virtuellem Speicher und wie es in Linux umgesetzt wird. Gehen Sie dabei auch auf folgende Punkte ein a) warum will man überhaupt virtuellen Speicher? b) Wie wird virtueller Speicher dem realen Speicher zugeordnet? c) Was geschieht wenn mehr virtueller Speicher verwendet wird als realer Hauptspeicher zur Verfügung steht?

- a) Warum virtuellen Speicher?

Da Programme mehr Speicherplatz benötigen als tatsächlich im Hauptspeicher verfügbar

- b) Wie zugeordnet?

Adressraum eines Programms wird in Einheiten unterteilt (Pages). Die Pages werden dem physischen Speicherbereich zugeordnet, wobei nicht alle Seiten zur selben Zeit im physischen Speicher vorhanden sein müssen.

- c) Mehr Virtueller Speicher verwendet als realer Hauptspeicher?

Wenn mehr virtueller Speicher verwendet wird als realer Hauptspeicher, dann tritt ein Page-Fault auf (bzw. TLB-Miss löst PageWalk aus), dadurch wird die Systemleistung beeinträchtigt

6

- b) Erklären Sie, was für einen Einfluss die Wahl eines Auslagerungs-Algorithmus auf die Leistung eines Systems hat und beschreiben Sie den Second Chance Algorithmus kurz.

Die Wahl eines Auslagerungs-Algorithmus kann Einfluss auf die Häufigkeit der Seitenfehler und damit wie oft ein Page Walk auftritt, welcher die Leistung eines Systems erheblich einschränkt.

Second Chance: Wie der Fifo-Algorithmus, vermeidet nur das Problem, dass Seiten ausgelagert werden, obwohl sie häufig benutzt werden. 5

- c) Erklären Sie die Funktion und Arbeitsweise des Buddy-Algorithmus. Wofür wird er verwendet und weshalb wurde er so entworfen wie er ist?

2^n aufeinanderfolgende, freie Pages werden zusammengefasst, dadurch ist eine sehr einfache Verwaltung von freien Pages möglich. Er wird zur Verwaltung des freien Speichers in Betriebssystemen eingesetzt.

Der Buddy-Allocator-Algorithmus teilt den frei verfügbaren Speicher in Blöcke (unterschiedliche Größen) und alloziert die Blöcke unter den Anforderungen der laufenden Prozesse. 5

Name _____

- d) Nennen und beschreiben Sie **jeweils** zwei Maßnahmen, die die temporale Lokalität und die spatiale Lokalität zur Optimierung der Performance beim Speichermanagement ausnutzen. Beschreiben Sie für beide Lokalitäten in einem Satz weshalb diese Mechanismen durch diese Lokalitäten funktioniert.

Temporale Lokalität (2 Maßnahmen):

Speicherzugriffe tendieren dazu, in kurzer Zeit wieder auf die gleiche oder ähnliche Speicherbereiche zuzugreifen

- Caching
- Prefetching

Spatiale Lokalität (2 Maßnahmen):

Speicherzugriffe tendieren dazu, räumlich wieder auf die gleiche oder ähnliche Speicherbereiche zuzugreifen

- Cache Line Prefetching
-

4

5. Kommunikation und Synchronisation. (11 Punkte)

- a) Was versteht man unter den Konzepten Deadlock und Livelock. Beschreiben Sie auch jeweils kurz eine problematische Situation, welche zum Auftreten des Problems führt.

Dead lock ist eine Situation, in der mehrere Prozesse aufeinander warten um benötigte Ressourcen frei zu geben. Keiner kann weiter arbeiten, da die anderen Prozesse die Ressourcen blockieren (→ Pattsituation)

Brennen einer Blu-ray-Disk: Prozess A reserviert zuerst Scanner, Prozess B zuerst Brenner beide warten darauf, dass der jeweils andere seine Ressourcen freigibt.

In der Livelock reagieren beide Prozesse aufeinander, aber sind in einer Endlosschleife gefangen.

5

- b) Was sind Signale unter Linux, wofür werden Sie verwendet und was ist zu tun, um sie verwenden zu können? Erklären Sie kurz.

Signale sind eine Methode zur Kommunikation zwischen Prozessen und Betriebssystem. Z.B. Interrupt-Signal um Prozesse zu unterbrechen (CTRL-C) oder Synchronisation SIGCHLD

Signal Handler im Programm implementieren und Signal verarbeiten

Mit kill vordefinierte Signale senden.

3

Name _____

- c) Betrachten Sie diesen Pseudocode als gegeben. Erklären Sie was getan werden muss, dass kein Philosoph mehr verhungert. Erklären Sie die von Ihnen gewählten Maßnahmen und markieren Sie in welchen Zeilen sie eingefügt werden müssen. Wie viele Philosophen können bei Ihrer Lösung gleichzeitig essen?

```
1 function philosoph(){
2   loop(while true){
3     wait(random)
4     esse()
5     wait(random)
6     denke()
7   }
8 }
9
10 function esse(){
11   nimmLinkeGabel()
12   nimmRechteGabel()
13   yummy()
14   legeLinkeGabelAb()
15   legeRechteGabelAb()
16 }
17
18 philosoph1 = thread(philosoph())
19 philosoph2 = thread(philosoph())
20 philosoph3 = thread(philosoph())
```

3

6. Virtualisierung

(10 Punkte)

- a) Was versteht man unter „Privilegierten Instruktionen“ und in welchem Zusammenhang stehen Sie zur Virtualisierbarkeit eines Systems?

4

- b) Nehmen Sie an, die Pagetable eines Gastsystems wäre durch das Host-System aus dem Speicher Hostsystems auf die Festplatte des Host-Systems ausgelagert worden. Beschreiben Sie was bei einem Pagewalk im Gastsystem nun insgesamt im Gesamtsystem geschieht und welche Auswirkungen dies auf die Performance des Gastsystems hat. Nehmen Sie dabei die ungünstigsten Bedingungen an.

6