

# Klausur “Betriebssysteme” – 5.2.2016

Die Dauer der Klausur beträgt 90 Minuten. Es sind keine Unterlagen und Hilfsmittel erlaubt. Bitte bearbeiten Sie die Aufgaben soweit wie möglich auf den Aufgabenblättern.

Nachname: \_\_\_\_\_

Vorname: \_\_\_\_\_

Matrikelnummer: \_\_\_\_\_

Klausurvoraussetzung erbracht im Semester (Zutreffendes ankreuzen):

☐ WS 15/16 ☐ SS 15 ☐ WS 14/15    Sonstiges

Unterschrift: \_\_\_\_\_

Aufgabe	Punktzahl maximal	Punktzahl erreicht
1	8	
2	14	
3	9	
4	10	
5	12	
6	18	
7	8	
8	11	
Summe	90	

## Aufgabe 1 (2+6 Punkte)

Punkte  von 8

Ein Prozessor benutzt zur Optimierung der Hauptspeicherzugriffe einen schnellen Cache-Speicher. Dieser ist organisatorisch in „Zeilen“ unterteilt. In jede Cache-Zeile passt ein 128 Byte großer zusammenhängender Bereich des Hauptspeichers. Die Anfangsadresse jedes dieser Bereiche ist immer ein Vielfaches der Zeilengröße 128.

- a) Nehmen Sie an, der Prozessor liest ein noch nicht im Cache vorhandenes 4 Byte großes Speicherwort mit der Adresse 264. Welche Speicherbytes stehen danach in der zugehörigen Cache-Zeile?
  
  
  
  
  
  
  
  
  
  
- b) Welche Informationen stehen außer dem Hauptspeicherinhalt sonst noch in dem Cache-Eintrag? Geben Sie die konkreten Werte dieser Informationen für das Beispiel an.

## Aufgabe 2 (4+4+6 Punkte)

Punkte  von 14

- a) Welches sind die einzelnen Schritte der Befehlsausführung bei einem Von-Neumann-Rechner?
  
  
  
  
  
  
  
  
  
  
- b) Was bedeutet „Pipelining“ im Zusammenhang mit der Ausführung von Maschinenbefehlen?

- c) Betrachten Sie folgende Maschinenbefehlssequenz eines RISC-Prozessors:

```
add $7,$8,$9    # Register 7 <--- Register 8 + Register 9
jr  $7           # Sprung zum Befehl, dessen Adresse in Register 7 steht
```

Inwieweit ist Pipelining bei dieser Befehlsfolge möglich? Treten Probleme auf? (Begründung)

### Aufgabe 3 (3+3+3 Punkte)

Punkte  von 9

- a) Ein blockierter Prozess wartet auf ein Ereignis  $E_1$ , z.B. eine Tastatureingabe. Typischerweise wird dieser Prozess irgendwann wieder aufgeweckt werden um dann später erneut auf ein anderes Ereignis  $E_2$  zu warten. Geben Sie ein Beispiel-Szenario an, in dem ein Prozess zunächst auf ein Ereignis  $E_1$  und danach auf ein anderes Ereignis  $E_2$  wartet, **ohne** dass er dazwischen aufgeweckt wird.
- b) Ein Betriebssystem unterscheidet bei Prozessen in Ausführung die Zustände „Running in user mode“ und „Running in kernel mode“. Welche Ereignisse können neben Systemaufrufen zu einem Übergang von dem ersten zu dem zweiten führen?
- c) Gibt es einen Übergang von „Running in user mode“ zu „Zombie“? (Begründung)

### Aufgabe 4 (3+5 Punkte)

Punkte  von 10

Betrachten Sie das nachfolgende C-Programm

```
#include <stdio.h>
#include <unistd.h>
int main(){
    pid_t pid;

    printf("%d, %d\n", getpid(), getppid());
    pid=fork();
    printf("%d, %d, %d\n", pid, getpid(), getppid());
}
```

Bei der Ausführung lautet die erste Ausgabezeile:

629, 621

- a) Von welchen Faktoren hängt der Rest der Ausgabe des Programms ab?
- b) Geben Sie die weiteren Ausgaben des Programms für verschiedene Ausführungs-Szenarien an?

## Aufgabe 5 (12 Punkte)

Punkte  von 12

Betrachten Sie folgenden Shell-Befehl:

```
od -x /bin/bash | grep ^00000 > pipe.out
```

Schreiben Sie ein C-Programm, das in vergleichbarer Weise die Programme `od` und `grep` in einer Pipeline kombiniert und die Ausgabe nach `pipe.out` umlenkt. Bei Fehlschlag von Systemaufrufen soll eine Fehlermeldung erscheinen, die einen Rückschluss auf die Fehlerursache erlaubt.

Die Aufgabe soll nicht an eine Shell (bzw. einen sonstigen Interpretierer) delegiert werden, wie z.B. mit `system("od -x /bin/bash | grep ^00000 > pipe.out")`

## Aufgabe 6 (2+4+4+4+4 Punkte)

Punkte  von 18

Ein System verwendet virtuellen Speicher mit Paging. Für die Berechnung der 32 Bit großen realen Adressen wird eine zweistufige Seitentabelle benutzt. Eine virtuelle Adresse ist 31 Bit groß und von der Form

$p_1$	$p_2$	$D$
-------	-------	-----

 mit folgender Aufteilung:

- $p_1$ : 5 Bit für den Seitentabellenindex der 1. Stufe
- $p_2$ : 10 Bit für den Seitentabellenindex der 2. Stufe
- $D$ : 16 Bit für die Distanz

Angenommen, die virtuelle Adresse  $v$  adressiert das 10. Byte der 3. Seite eines Prozesses, die im 7. Rahmen des Hauptspeichers steht.

- a) Wie groß ist ein Seitenrahmen?
- b) Geben Sie  $v$  und  $r$  hexadezimal an.  
 $v = 0x$   $r = 0x$
- c) Geben Sie an, wie der TLB-Eintrag zu  $v$  und  $r$  aussieht:
- d) Wenn der virtuelle Adressraum eines Prozesses 512 Seiten groß ist, wieviele Seitentabellen gibt es dann für diesen Prozess in jeder Stufe?  
Anzahl der Seitentabellen in der 1. Stufe: in der 2. Stufe:
- e) Was wäre ein geeigneter Speicherort für die Seitentabelle der ersten Stufe? (Begründung)

## Aufgabe 7 ( Punkte)

Punkte  von 8

Ein Scheduler richtet sich bei der CPU-Vergabe nach den Prioritäten der Threads: Ein Thread darf nicht ausgeführt werden, wenn ein anderer mit höherer Priorität auf CPU-Zuteilung wartet. Die Prioritäten werden dynamisch berechnet und können jederzeit durch Benutzereingriffe geändert werden.

Welche Umstände können hier zu einem Kontextwechsel führen? Ergänzen Sie die Liste!

1. Der ausführende Thread muss auf irgendein Ereignis warten und blockiert.
2. \_\_\_\_\_
3. \_\_\_\_\_
4. \_\_\_\_\_
5. \_\_\_\_\_
6. \_\_\_\_\_

## Aufgabe 8 (3+4+4 Punkte)

Punkte  von 11

a) Wozu enthält bei einem UNIX-Dateisystem jeder Inode einen Referenzzähler ?

b) Ein C-Programm, das auf einer UNIX-Plattform läuft, versucht eine Datei zum Lesen zu öffnen:

```
fd = open("/tmp/datei.txt", O_RDONLY, 0);
```

Der Systemkern muss dazu zunächst den Pfad Schritt für Schritt verarbeiten und dabei unter anderem diverse Zugriffsrechtsprüfungen durchführen. Geben Sie ausgehend vom I-Node des Wurzelverzeichnis alle Teilschritte dieser Pfadverarbeitung an.

c) Nach dem erfolgreichen Öffnen gemäß (b) führt das Programm zwei Lesezugriffe aus:

```
read(fd, buffer, 4);  
read(fd, buffer, 4);
```

Obwohl im `read`-Aufruf keine Leseposition angegeben ist, werden im ersten Aufruf die ersten 4 Byte der Datei gelesen und im zweiten Aufruf die dahinter stehenden nächsten 4 Byte. Geben Sie mit Hilfe einer Skizze an, welche Datenstrukturen für geöffnete Dateien hier im Spiel sind und wo die aktuelle Leseposition abgespeichert ist.