

Klausur “Betriebssysteme I” – 18.3.2011

Bitte bearbeiten Sie die Aufgaben auf den Aufgabenblättern.
Die Benutzung von Unterlagen oder Hilfsmitteln ist nicht erlaubt.
Die Bearbeitungszeit beträgt 90 Minuten.

Nachname: _____

Vorname: _____

Studiengang/Prüfungsordnung: _____

Matrikelnummer: _____

Klausurvoraussetzungen (Hausübungen usw.) aus Semester:
(Zutreffendes ankreuzen)

☐ WS 10/11 ☐ SS 10 ☐ WS 09/10 ☐ Sonstiges

Unterschrift: _____

Aufgabe	Punktzahl maximal	Punktzahl erreicht
1	7	
2	4	
3	4	
4	4	
5	6	
6	5	
7	6	
8	4	
9	4	
10	4	
Summe	48	

Aufgabe 1 (3+1+3 Punkte)

Punkte von 7

Betrachten Sie für einen Prozess oder Thread den Zustandsübergang „ausführend“ → „bereit“.

- a) Welche Ereignisse können zu diesem Zustandsübergang führen?

- b) Wie heißt die Betriebssystemkomponente, die über diesen Zustandsübergang entscheidet?

- c) Beschreiben Sie in Stichworten, wie der Systemkern diesen Übergang konkret implementiert.

Aufgabe 2 (4 Punkte)

Punkte von 4

Sie testen unter UNIX ein selbstentwickeltes Programm. Der Prozess wird mit der Meldung „Division durch Null“ terminiert. Sie stellen fest, dass ihr Programm eine solche Division versucht hat. Allerdings haben Sie für diesen Fall weder die Ausgabe der Fehlermeldung noch den Abbruch der Ausführung programmiert.

Offenbar führt der Prozessor Befehle aus, die nicht im Programm stehen! Wie kommt es dazu? Beschreiben Sie den Mechanismus!

Was ist daran UNIX-spezifisch?

Aufgabe 3 (4 Punkte)

Punkte von 4

Welche möglichen Ausgaben hat das nachfolgende C-Programm. Das Programm wird von einer Shell mit PID 30 aufgerufen, der Hauptprozess erhält PID 38 und der Subprozess PID 42.

```
#include ...
...
int main() {
    int i=0;
    printf("1. %d\n", getpid());
    i=fork();
    printf("2. %d - %d - %d \n", getpid(), getppid(), i);
    waitpid(i,NULL,0);
    if (i==0)
        printf("3. %d\n", getpid());
    exit(0);
}
```

Aufgabe 4 (4 Punkte)

Punkte von 4

Eine Autobrücke trägt nur eine bestimmte Last, so dass beim Befahren eine wichtige Einschränkung zu beachten ist: Wenn ein LKW auf die Brücke fährt, darf sich kein anderes Auto auf der Brücke befinden. Beliebige viele PKW dürfen gleichzeitig die Brücke befahren.

Wenn man dies in einem Software-Verkehrsmodell abbilden will, benötigt man bestimmte Warteschlangenoperationen. Welcher Ihnen bekannte Synchronisationsmechanismus bietet sich zur Lösung an? Wie würden Sie diesen Synchronisationsmechanismus hier konkret verwenden?

Aufgabe 5 (6 Punkte)

Punkte von 6

Betrachten Sie folgendes Programm. Beachten Sie, dass der Kindprozess *anzahl* Bytes schreibt, der Elternprozess aber nur 1 Byte liest.

```
const int anzahl=...; // zwischen 1 und 100000
char daten[anzahl];
main() {
    int pipefd[2];
    pipe(pipefd);

    switch(fork()){
        case -1: perror("fork-Fehler"); exit(1);
        case 0: write(pipefd[1], daten, anzahl);
                exit(0);
    }

    // Elternprozess
    read(pipefd[0], daten, 1);
    waitpid(-1,NULL,0);
}
```

Kennzeichnen Sie die nachfolgenden Aussagen jeweils als „wahr“ oder „falsch“:

- a) Bei der Ausführung kommt es immer zu einer Verklemmung.
- b) Es kommt in keinem Fall zu einer Verklemmung
- c) Ob es zu einer Verklemmung kommt, hängt von *anzahl* ab.
- d) Ob es zu einer Verklemmung kommt, hängt davon ab, ob der Elternprozess oder der Kindprozess den Prozessor zuerst bekommt.
- e) Das Programmverhalten ändert sich, wenn der Kindprozess als erstes `pipefd[0]` schließt.
- f) Wenn der *waitpid*-Aufruf entfernt wird, kann es in keinem Fall zu einer Verklemmung kommen.

Aufgabe 6 (5 Punkte)

Punkte von 5

Ein Programm soll die Zeilenanzahl der Datei „dat.txt“ ausgeben. Das Programm besteht aus zwei Prozessen. Der Hauptprozess kopiert den Datei-Inhalt in eine Pipe. Der Subprozess delegiert die Ausgabe der Zeilenanzahl an das externe Programm „wc“ (Argument „-l“). Dessen Standardeingabe wird vor dem Aufruf an die Pipe gebunden.

Vervollständigen Sie dazu den nachfolgenden Quelltext. Benutzen Sie nur die folgenden Systemaufrufe: *fork*, *execlp*, *open*, *read*, *write*, *pipe*, *close*, *dup2*. Verzichten Sie auf Fehlerbehandlung. Achten Sie auf Verklemmungsfreiheit.

```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <stdlib.h>
#include <sys/stat.h>
#include <fcntl.h>

int main() {

    int fd, pfd[2];
    char c;

    pipe(pfd);

    switch( fork() ) {
    case 0:

        }

        fd=open("dat.txt", O_RDONLY, 0);
```

Aufgabe 7 (1+2+2+1 Punkte)

Punkte von 6

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

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

mit folgender Aufteilung:

- p_1 : 9 Bit für die Seitentabelle der 1. Stufe
- p_2 : 11 Bit für die Seitentabelle der 2. Stufe
- D : 12 Bit für die Distanz

- a) Wie groß ist eine Seite des virtuellen Adressraums?
- b) Wie groß ist der reale Adressraum, der durch einen Seitentableneintrag in der 1. Stufe repräsentiert wird?
- c) Geben Sie eine Skizze an, aus der die Verwendung von p_1 , p_2 und D bei der Bestimmung der realen Adresse hervorgeht.

- d) Wie wird in diesem System die Distanz der realen Adresse zum Seitenrahmen berechnet und welche Rolle spielt der TLB dabei?

Aufgabe 8 (4 Punkte)

Punkte von 4

In einem Multiprozessorsystem führen zwei Threads gleichzeitig **asynchrone** Plattenzugriffe aus. Der Systemkern blockiert dabei die Threads während des Plattenzugriffs nicht, sondern erzeugt nur je einen neuen Eintrag in der Auftragsliste für den Plattencontroller. Ist gegenseitiger Ausschluss beim Zugriff auf diese Liste nötig? Wenn ja, ist hier ein Spinlock oder ein Mutex besser geeignet? Begründen Sie die Antworten.

Aufgabe 9 (4 Punkte)

Punkte von 4

In einem System mit Paging steht ein nur 3 Rahmen großer Hauptspeicher zur Verfügung. Geben Sie für die Auslagerungsstrategien FIFO und LRU die Speicherbelegung und die Anzahl der Seitenfehler nach folgenden Seitenzugriffen an:

8 5 3 2 5 6 2 3 4 2

FIFO

Zugriffe	8	5	3	2	5	6	2	3	4	2
Rahmen 1	8	8	8							
Rahmen 2		5	5							
Rahmen 3			3							

Seitenfehler:

LRU

Zugriffe	8	5	3	2	5	6	2	3	4	2
Rahmen 1	8	8	8							
Rahmen 2		5	5							
Rahmen 3			3							

Seitenfehler:

Aufgabe 10 (4 Punkte)

Punkte von 4

Das Root-Verzeichnis eines FAT-Dateisystems ist nicht mehr lesbar.

Inwieweit und mit welchem Ergebnis lassen sich die Dateien in folgendem Beispiel anhand der FAT rekonstruieren? (ein mit „—“ dargestellter FAT-Eintrag steht für das letzte Cluster einer Datei).

0	—
1	frei
2	—
3	—
4	7
5	2
6	frei
7	3
8	frei
9	5
10	frei
11	4