

Aufgabe 1.1: Einfachauswahl-Fragen (3 Punkte)

Bei den Multiple-Choice-Fragen in dieser Aufgabe ist jeweils nur **eine** richtige Antwort eindeutig anzukreuzen. Auf die richtige Antwort gibt es die angegebene Punktzahl.

Wollen Sie eine Multiple-Choice-Antwort korrigieren, streichen Sie bitte die falsche Antwort mit drei waagrechten Strichen durch (~~☒~~) und kreuzen die richtige an.

Lesen Sie die Frage genau, bevor Sie antworten.

a) Gegeben seien die folgenden Präprozessor-Makros:

```
#define ADD(a, b) a + b
```

```
#define DIV(a, b) a / b
```

Was ist das Ergebnis des folgenden Ausdrucks?

```
3 * DIV(ADD(4, 8), 2)
```

- 10
 24
 16
 18

1,5 P.

b) Was versteht man unter Virtuellem Speicher?

- Speicher, der nur im Betriebssystem sichtbar ist, jedoch nicht für einen Anwendungsprozess.
- Speicher, der einem Prozess durch entsprechende Hardware (MMU) und durch Ein- und Auslagern von Speicherbereichen vorgespiegelt wird, aber möglicherweise größer als der verfügbare physikalische Hauptspeicher ist.
- Unter einem Virtuellen Speicher versteht man einen physikalischen Adressraum, dessen Adressen durch eine MMU vor dem Zugriff auf logische Adressen umgesetzt werden.
- Virtueller Speicher kann dynamisch zur Laufzeit von einem Programm erzeugt werden (Funktion `valloc(3)`).

1,5 P.

Aufgabe 1.2: Mehrfachauswahl-Fragen (3 Punkte)

Bei den Multiple-Choice-Fragen in dieser Aufgabe sind jeweils m Aussagen angegeben, n ($0 \leq n \leq m$) Aussagen davon sind richtig. Kreuzen Sie **alle richtigen** Aussagen an. Jede korrekte Antwort in einer Teilaufgabe gibt einen halben Punkt, jede falsche Antwort einen halben Minuspunkt. Eine Teilaufgabe wird minimal mit 0 Punkten gewertet, d. h. falsche Antworten wirken sich nicht auf andere Teilaufgaben aus.

Wollen Sie eine falsch angekreuzte Antwort korrigieren, streichen Sie bitte das Kreuz mit drei waagrechten Strichen durch (~~☒~~).

Lesen Sie die Frage genau, bevor Sie antworten.

a) Welche der folgenden Aussagen zum Thema Dateisysteme sind richtig?

- In einem hierarchisch organisierten Namensraum dürfen gleiche Namen in unterschiedlichen Kontexten enthalten sein.
- UNIX-Betriebssysteme unterstützen keine Dateisysteme mit hierarchischem Namensraum.
- Symbolische Links können nur auf Dateien innerhalb des selben Dateisystems verweisen.
- Es ist möglich, einen symbolischen Link mit Verweis auf eine nicht-existierende Datei anzulegen.
- Ein Inode enthält Metadaten über eine Datei: Größe, Eigentümer, Zugriffsrechte, Dateiname usw.
- In einem UNIX-Dateisystem ist es möglich, dass derselbe Inode unter mehreren Dateinamen erreichbar ist.

3 Punkte

Aufgabe 2: (15 Punkte)

Sie dürfen diese Seite und die Manual-Seite am Ende der Klausur zur besseren Übersicht bei der Programmierung heraustrennen!

Schreiben Sie eine Funktion

```
int getSortedEntries(struct Entry entries[],
                    unsigned int maxEntries);
```

die alle Einträge des aktuellen Arbeitsverzeichnisses nach Größe absteigend sortiert in ein Array von Tupeln der Form (Name, Größe) schreibt. Einträge mit identischer Größe sollen aufsteigend in alphabetischer Reihenfolge geordnet sein.

Das Zielarray wird vom Aufrufer angelegt und zusammen mit seiner Größe an getSortedEntries() übergeben. Falls das Array komplett gefüllt ist und das Verzeichnis noch weitere Einträge enthält, werden diese ignoriert. Alle bis dahin eingelesenen Tupel werden aber dennoch sortiert.

Über den Rückgabewert wird signalisiert, welcher Fall eingetreten ist: War das Array groß genug, liefert die Funktion die Anzahl der eingelesenen Einträge zurück - war es zu klein, wird der negative Wert -maxEntries zurückgegeben. Falls während der Ausführung ein Fehler auftritt, wird die Bearbeitung abgebrochen und der Wert 0 zurückgeliefert. Es sollen keine Fehlermeldungen ausgegeben werden.

Implementieren Sie außerdem eine Hilfsfunktion zum Vergleichen zweier Tupel und greifen Sie beim Sortieren auf diese Funktion zurück.

```
#include <dirent.h>
#include <errno.h>
#include <limits.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/stat.h>
```

```
// Tupel-Struktur
struct Entry {
    char name[NAME_MAX + 1];
    off_t size;
};
```

```
// Hilfsfunktion fuer qsort()
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

```
// Funktion getSortedEntries()
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Aufgabe 3: (9 Punkte)

- a) Erläutern Sie das Konzept "Semaphor". Welche Operationen sind auf Semaphoren definiert und was tun diese Operationen? (6 Punkte)

- b) Skizzieren Sie in Programmiersprachen-ähnlicher Form, wie mit Hilfe eines zählenden Semaphors das folgende Szenario korrekt synchronisiert werden kann: Ein Hauptthread soll so lange warten, bis 5 Arbeiter-Threads ihre Arbeit (Ausführung einer Funktion `doWork()`) erledigt haben. (3 Punkte)

Hauptthread:

```
sem = sem_init(0);
startWorkerThreads();
```

Arbeiter-Thread:

```
doWork();
```
