

Aufgabe 1: Ankreuzfragen (7 Punkte)**1) Einfachauswahlfragen (4 Punkte)**

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

Wollen Sie eine 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) Man unterscheidet zwischen Traps und Interrupts. Welche Aussage ist richtig?

2 Punkte

- ☐ Traps können nicht durch Speicherzugriffe ausgelöst werden.
- ☐ Bei der mehrfachen Ausführung eines unveränderten Programmes mit den selben Eingabedaten treten Interrupts immer an den selben Stellen auf.
- ☐ Traps stehen immer in ursächlichem Zusammenhang zu der Ausführung eines Maschinenbefehls.
- ☐ Ein Trap wird immer durch das Programm behandelt, welches den Trap ausgelöst hat, Interrupts werden hingegen immer durch das Betriebssystem behandelt.

b) Welche Aussage zu Semaphoren ist richtig?

2 Punkte

- ☐ Die Up-Operation eines Semaphors kann ausschließlich von einem Thread aufgerufen werden, der zuvor mindestens eine Down-Operation auf dem selben Semaphor aufgerufen hat.
- ☐ Die Down-Operation eines Semaphors erhöht den Wert des Semaphors um 1 und deblockiert gegebenenfalls wartende Prozesse.
- ☐ Ein Semaphor kann nur zur Signalisierung von Ereignissen, nicht jedoch zum Erreichen gegenseitigen Ausschlusses verwendet werden.
- ☐ Die Up-Operation eines Semaphors erhöht den Wert des Semaphors um 1 und deblockiert gegebenenfalls wartende Prozesse.

2) Mehrfachauswahlfragen (3 Punkte)

Bei den Mehrfachauswahlfragen in dieser Aufgabe sind jeweils m Aussagen angegeben, davon sind n ($0 \leq n \leq m$) Aussagen richtig. Kreuzen Sie alle richtigen Aussagen an.

Jede korrekte Antwort in einer Teilaufgabe gibt einen Punkt, jede falsche Antwort einen 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) Man unterscheidet die Begriffe Programm und Prozess. Welche der folgenden Aussagen zu diesem Themengebiet ist richtig?

3 Punkte

- ☐ Ein Prozess ist ein Programm in Ausführung - ein Prozess kann während seiner Lebenszeit aber auch mehrere verschiedene Programme ausführen.
- ☐ Der Prozess ist der statische Teil (Rechte, Speicher, etc.), das Programm der aktive Teil (Programnzähler, Register, Stack).
- ☐ Mit Hilfe des Systemaufrufs `execve(2)` (bzw. der Bibliotheksfunktion `exec(3)`) wird das bestehende Programm im aktuell laufenden Prozess ersetzt.
- ☐ Der C-Präprozessor übersetzt mehrere C-Quelldateien in Module die zu einem Programm gebunden werden können.
- ☐ Ein Programm kann durch mehrere Prozesse gleichzeitig ausgeführt werden.
- ☐ Ein Prozess kann mit Hilfe von Threads mehrere Programme gleichzeitig ausführen.

Aufgabe 2: pdfsize (15 Punkte)

Sie dürfen diese Seite zur besseren Übersicht bei der Programmierung heraustrennen!

Schreiben Sie ein Programm `pdfsize`, welches einen Verzeichnispfad als Argument erhält und daraufhin alle `.pdf`-Dateien in dem übergebenen Verzeichnis inklusive deren Dateigröße ausgibt. Ermitteln Sie dazu alle Dateien mit der Dateiendung `.pdf` und sortieren Sie diese aufsteigend nach deren Größe (kleinere Dateien zuerst). Das Programm soll dabei nicht in Unterverzeichnisse absteigen und auch symbolischen Verknüpfungen nicht folgen. Abschließend soll die Gesamtgröße aller gefundenen `.pdf`-Dateien ausgegeben werden. Hinweis: Die Fehlerbehandlung von Ausgabeoperationen ist nicht nötig. Alle anderen Fehler sollen sinnvoll behandelt werden.

Beispielaufruf von `pdfsize`:

```
$ ./pdfsize klausur/
klausur/man.pdf: 43607
klausur/2025w-SP-MKlausur.pdf: 229590
klausur/2025w-SP-MKlausur_ml.pdf: 240367
Gesamtgroesse: 513564
```

Implementieren Sie die folgenden Funktionen:

int main(int argc, char **argv)

Ermittelt alle `.pdf`-Dateien mittels `scandir(3p)` und gibt diese aufsteigend sortiert nach Dateigröße aus. Die Ausgabe beinhaltet jeweils den Dateipfad und die Dateigröße in Byte. **Verwenden Sie zum Ermitteln der Dateigröße die vorgegebene Funktion `get_filesize`.** Abschließend wird die Gesamtgröße aller gefundenen `.pdf`-Dateien (in Byte) ausgegeben.

int filter(const struct dirent *ent)

Filterfunktion für `scandir`. Filtert die Verzeichniseinträge nach `.pdf`-Dateien mittels `fnmatch(3p)`. Es sollen nur reguläre Dateien berücksichtigt werden, symbolischen Verknüpfungen soll nicht gefolgt werden.

int sizesort(const struct dirent **a, const struct dirent **b)

Sortierfunktion für `scandir`.

```
#include <...>
```

```
// Per Kommandozeile übergebener Pfad
```

```
static char *path;
```

```
static void die(const char *msg) {
    perror(msg);
    exit(EXIT_FAILURE);
}
```

```
static int get_filesize(const struct dirent *entry) {
    // Gibt die Dateigröße in Byte zurück; kann nicht fehlschlagen
}
```

```
int main(int argc, char **argv) {
```

```
    // Argumente überprüfen
```

```
    // Pfad global merken
```

```
    // Dateien im Verzeichnis ermitteln
```

```
    // Ermittelte Dateien und Gesamtgröße ausgeben
```

```
    return EXIT_SUCCESS;
```

```
}
```

☐☐☐**M:**

}



F:

}

S:

Aufgabe 3: Programmieren und Speicher (8 Punkte)

1) Folgendes Programm enthält Programmierfehler. Nennen Sie diese und beschreiben Sie die möglichen Auswirkungen. (3 Punkte)

```
1 int main(void) {  
2     // Reserviere Speicher für 64 Integer  
3     int *p = malloc(64);  
4     for (int i = 0; i < 64; i++) {  
5         p[i] = i;  
6     }  
7     free(p);  
8     return 0;  
9 }
```

2) Schreiben Sie den vorherigen Programmcode so um, dass nur Speicher auf dem Stack verwendet wird. Wie unterscheidet sich der Parameter eines malloc-Aufrufs von der Größendefinition eines Arrays? (3 Punkte)

```
int main(void) {
```

```
}
```

Erklärung:

3) Nennen Sie den Bereich, in dem der von malloc allokierte Speicher verwaltet wird, sowie einen Nachteil im Vergleich zur Nutzung von Stack-Speicher. (2 Punkte)