## Aufgabe 1.1: Einfachauswahl-Fragen (4 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) Was versteht man unter Adressraumschutz durch Eingrenzung? 2 Punkte

- ❏ Der Compiler grenzt beim Erzeugen des Codes die Adresszugriffe auf einen bestimmten Bereich ein.
- ❏ Zur Laufzeit eines Programms ist durch Begrenzungsregister festgelegt, auf welchen Bereich des physikalischen Speichers das Programm zugreifen darf.
- ❏ Beim Laden eines Programms prüft das Betriebssystem, ob alle Speicherzugriffe im gültigen Bereich liegen.
- ❏ Die Zugriffe eines Programms werden zur Laufzeit auf die aktuell gültigen Segmente eingegrenzt.

b) Der Speicher eines UNIX-Prozesses ist in Text-, Daten- und Stack-(Stapel-)Segment untergliedert. Welche Aussage zur Platzierung von Daten in diesen Segmenten ist richtig? 2 Punkte

- ❏ Alle lokalen Variablen werden im Stack-Segment abgelegt.
- ❏ Der Code von Funktionen wird zusammen mit den Variablen der Funktion im Stack-Segment abgelegt.
- ❏ Variablen der Speicherklasse *static* liegen im Daten-Segment.
- ❏ Bei einem malloc-Aufruf wird das Stack-Segment dynamisch erweitert.





## 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) Man unterschiedet zwei Kategorien von Ausnahmesituationen bei einer Programmausführung: Traps und Interrupts. Welche der folgenden Aussagen sind zutreffend? 3 Punkte

- ❍ Ein Systemaufruf führt zu einem Trap.
- ❍ Wenn ein Interrupt ein schwerwiegendes Ereignis signalisiert, wird das unterbrochene Programm im Rahmen der Interruptbearbeitung abgebrochen.
- ❍ Nach der Behandlung eines Traps kann das unterbrochene Programm gegebenenfalls fortgesetzt werden.
- ❍ Ein Programm darf im Rahmen einer Trapbehandlung abgebrochen werden.
- ❍ Die CPU sichert bei einem Interrupt einen Teil des Prozessorzustands.
- ❍ Bei einem Trap wird der gerade in Bearbeitung befindliche Befehl immer noch vollständig zu Ende bearbeitet, bevor mit der Trapbehandlung begonnen wird.

## Aufgabe 2: (15 Punkte)

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

Implementieren Sie eine Funktion **`newer`**, die Dateien in einem geöffneten Verzeichnis findet, die neuer sind als eine Referenzdatei.

Hierzu wird das Änderungsdatum der Referenzdatei (Parameter **`refFile`**) ermittelt und anschließend das Verzeichnis (Parameter **`dir`**) nach regulären Dateien durchsucht, deren Änderungsdatum jünger ist.

Ist eine Datei neuer, so wird ein Zeiger auf den Dateinamen in das Feld **`files`** kopiert. In diesem können maximal **`maxFiles`** Zeiger abgelegt werden, danach wird die Bearbeitung des Verzeichnisses abgebrochen.

Die Funktion gibt die Anzahl der im Feld **`files`** abgelegten Zeiger zurück. Tritt ein Fehler auf, liefert sie die Anzahl der bereits abgelegten Zeiger als negative Zahl.

Schnittstelle:

```
int newer(DIR *dir, char *refFile, char *files[], int maxFiles);
```

Hinweise:

– Die Funktion soll keine Meldungen ausgeben, auch keine Fehlermeldungen.

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





```
/* Funktion newer */
```

**Aufgabe 3: (9 Punkte)**

Beschreiben Sie die unterschiedlichen Arten ("Gewichtsklassen") von Prozessen/ Threads.