

Aufgabe 1: (6 Punkte)

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

Wollen Sie eine Multiple-Choice-Antwort korrigieren, kreisen sie bitte die falsche Antwort ein und kreuzen die richtige an.

Lesen Sie die Frage genau, bevor Sie antworten.

- a) Ein Betriebssystem setzt logische Adressräume auf der Basis von Segmentierung ein. Welche Aussage ist richtig? 2 Punkte
- Die Bindung von Programm- an Arbeitsspeicheradressen erfolgt zur Ladezeit des Programms.
 - Die Segmentierung schränkt den logischen Adressraum derart ein, dass nur auf gültige Speicheradressen erfolgreich zugegriffen werden kann.
 - Adressraumschutz durch Segmentierung erfordert keine Hardwareunterstützung.
 - Segmente können verschiedene Länge haben. Die Einhaltung der Längenbegrenzung wird vom C-Compiler überprüft.
- b) Nehmen Sie an, der Ihnen bekannte Systemaufruf `stat(2)` wäre analog zu der Funktion `readdir(3)` mit folgender Schnittstelle implementiert:

```
struct stat *stat(const char *path);
```

 Welche Aussage ist richtig? 2 Punkte
- Der Systemaufruf liefert einen Zeiger zurück, über den die aufrufende Funktion direkt auf eine Datenstruktur zugreifen kann, die die Dateiattribute enthält.
 - Solch eine Schnittstelle ist nicht schön, da dadurch die aufrufende Funktion auf internen Speicher des Betriebssystems zugreifen könnte.
 - Der Aufrufer muss sicherstellen, dass er den zurückgelieferten Speicher mit `free(3)` wieder freigibt, wenn er die Dateiattribute nicht mehr benötigt.
 - Ein Zugriff über den zurückgelieferten Zeiger liefert völlig zufällige Ergebnisse oder einen Segmentation fault.
- c) Namensräume dienen u. a. der Organisation von Dateisystemen. Welche Aussage ist richtig? 2 Punkte
- Flache Namensräume sind besonders einfach implementierbar und damit vor allem für Mehrbenutzersysteme gut geeignet.
 - Flache Namensräume erlauben pro Benutzer nur einen Kontext.
 - In einem hierarchisch organisierten Namensraum dürfen gleiche Namen in unterschiedlichen Kontexten enthalten sein.
 - Hierarchische Namensräume werden erzeugt, indem man in einem Kontext symbolische Verweise auf Dateien einträgt.

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 **execute**, die eine Zeichenkette (Parameter `line`) mittels `execvp()` ausführt und den Exit-Status des gestarteten Programmes zurückliefert.

Dazu soll die übergebene Zeile an Tabulatoren und Leerzeichen geteilt und so in Kommandoname und Argumente zerlegt werden. Anschließend wird das Kommando mit den Argumenten in einem neu erzeugten Prozess ausgeführt. Beendet sich dieses Programm selbst, soll dessen Exit-Status zurückgegeben werden. Ansonsten wird `-2` zurückgegeben.

Schnittstelle:

```
int execute(const char *line);
```

Rückgabewerte im Fehlerfall:

- 1: bei der Bearbeitung trat ein Fehler auf
- 2: das Kind lieferte keinen Exit-Status

Hinweise:

- Es ist nicht notwendig, die Funktionen `malloc()` und `realloc()` zu verwenden.
- Die Funktion soll keine Meldungen ausgeben, auch keine Fehlermeldungen.

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

