

**Aufgabe 1: (8 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) Welche Aussage zum Thema symbolische Links und Hard-Links ist **falsch**? 2 Punkte
- Jeder symbolische Link verfügt über einen eigenen Inode.
  - Symbolische Links können auf nicht existierende Dateien verweisen.
  - Symbolische Links sind nur innerhalb desselben Namensraumes möglich.
  - Jedes Verzeichnis ist mit mindestens zwei Hard-Links im Dateisystem eingetragen.
- b) Man unterscheidet Traps und Interrupts. Welche Aussage ist **falsch**? 2 Punkte
- Für einige Maschineninstruktionen kann sicher vorhergesagt werden, dass durch ihre Ausführung niemals ein Trap ausgelöst wird.
  - Für einige Maschineninstruktionen kann nicht sicher vorhergesagt werden, ob durch ihre Ausführung ein Trap ausgelöst wird oder nicht.
  - Interrupts dürfen nicht nach dem Beendigungsmodell behandelt werden, da zwischen dem aufgetretenen Interrupt und dem aktuell ausgeführten Prozess u.U. kein kausaler Zusammenhang besteht.
  - Der Arbeitsspeicher generiert beim Erkennen eines ungültigen Speicherzugriffs einen Trap.

- c) Welche Aussage zu den verschiedenen Gewichtsklassen von Prozessen **trifft zu**? 2 Punkte
- Federgewichtige Prozesse laufen im Gegensatz zu den leichtgewichtigen Prozessen im gleichen Adressraum und sind daher ressourcenschonender.
  - Blockiert ein Benutzerfaden eines federgewichtigen Prozesses in einem Systemaufruf wie **read(2)**, so werden auch alle anderen Benutzerfäden des federgewichtigen Prozesses blockiert und müssen auf die Rückkehr des Systemaufrufs warten.
  - Die Einplanung und Einlastung von Benutzerfäden leichtgewichtiger Prozesse findet immer ohne Wissen des Betriebssystems statt und erfordert daher entsprechende Infrastruktursoftware innerhalb der Anwendung selbst.
  - Schwergewichtige Prozesse sind die einzige Klasse von Prozessen, die auf einem Multiprozessorsystem echt parallel ausgeführt werden kann, da nur hier jeder Benutzerfaden einem eigenen Kernfaden zugeordnet ist.
- d) Welche Aussage im Bezug auf Adressräume **trifft zu**? 2 Punkte
- Die Adressumsetzung von logischen auf physikalische Adressen erfolgt zur Ladezeit durch eine spezielle Hardwareeinheit (MMU).
  - In einem physikalischen Adressraum sind immer alle Adressen gültig.
  - Unter virtuellem Speicher versteht man die ungültigen Adressen eines logischen Adressraums.
  - Bei Speicherschutz durch Eingrenzung laufen die Programme im physikalischen Adressraum.

**Aufgabe 2: (15 Punkte)**

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

Programmieren Sie eine Funktion **exec\_path**, die ein Programm (Parameter prg) in einem Pfad (gegeben durch eine NULL-terminierte Liste von Verzeichnissen, Parameter paths) sucht.

Die erste hierbei gefundene reguläre Datei, die für **jedermann ausführbar** ist (bzw. ein symbolischer Link auf eine solche Datei), wird als Programm mit den bereitgestellten Argumenten (Parameter argv) in den aktuellen Prozess geladen.

Schnittstelle:

```
int exec_path(char *paths[], char *prg, char *argv[]);
```

Rückgabewert:

- 1: kein passendes Programm gefunden
- 2: beim Versuch, ein passendes Programm zu laden, trat ein Fehler auf

Anmerkungen:

- Sie können davon ausgehen, dass ein Pfadname niemals länger als PATH\_MAX Zeichen ist.
- Pfad-Konkatenationen können alternativ mit strcpy(3) und strcat(3) oder mit snprintf(3) durchgeführt werden!
- Zum Test, ob eine reguläre Datei für jedermann ausführbar ist, können Sie das vorgegebene Makro IS\_EXEC(ST\_MODE) verwenden. Dieses erwartet als Parameter den Wert des st\_mode-Feldes der stat-Struktur.

```
/* includes */
#include <stdio.h>
#include <stdlib.h>
#include <sys/stat.h>
#include <unistd.h>
#include <limits.h>

#define EXECALL (S_IXUSR|S_IXGRP|S_IXOTH)
#define IS_EXEC(ST_MODE) (EXECALL==(ST_MODE & EXECALL))
```

**/\* Funktion exec\_path \*/**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

**Aufgabe 3: (7 Punkte)**

Der logische Adressraum eines Prozesses unter UNIX/Linux ist in Segmente aufgeteilt. Nennen Sie diese und beschreiben Sie kurz, welche Daten darin jeweils angelegt werden.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....