

**Aufgabe 1.1: Einfachauswahl-Fragen (18 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) Welche der folgenden Aussagen zum Thema Adressräume ist richtig? 2 Punkte
- Der virtuelle Adressraum eines Prozesses kann nie größer sein als der physikalisch vorhandene Arbeitsspeicher.
  - Die maximale Größe des virtuellen Adressraums kann unabhängig von der verwendeten Hardware frei gewählt werden.
  - Virtuelle Adressräume sind Voraussetzung für die Realisierung logischer Adressräume.
  - Logische Adressräume bieten Schutz vor Berechnungsfehlern.
- 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.
  - Variablen der Speicherklasse *static* liegen im Daten-Segment.
  - Bei einem malloc-Aufruf wird das Stack-Segment dynamisch erweitert.
  - Dynamisch allozierte Zeichenketten liegen im Text-Segment.
- c) Man unterscheidet die Begriffe Programm und Prozess. Welche der folgenden Aussagen zu diesem Themengebiet ist richtig? 2 Punkte
- Das Programm ist der statische Teil (Rechte, Speicher, etc.), der Prozess der aktive Teil (Programmzähler, Register, Stack).
  - Der Compiler erzeugt aus mehreren Programmteilen (Module) einen Prozess.
  - Ein Programm kann durch mehrere Prozesse gleichzeitig ausgeführt werden.
  - Ein Prozess kann mit Hilfe von Threads mehrere Programme gleichzeitig ausführen.

- d) Welche Aussage über `exec()` ist richtig? 2 Punkte
- Das im aktuellen Prozess laufende Programm wird durch das angegebene Programm ersetzt.
  - Der an `exec()` übergebene Funktionszeiger wird durch einen neuen Thread im aktuellen Prozess ausgeführt.
  - `exec()` erzeugt einen neuen Kind-Prozess und startet darin das angegebene Programm.
  - Dem Vater-Prozess wird die Prozess-ID des Kind-Prozesses zurückgeliefert.
- e) Bei der Behandlung von Ausnahmen (Traps oder Interrupts) unterscheidet man zwei Bearbeitungsmodelle. Welche Aussage hierzu ist richtig? 2 Punkte
- Das Wiederaufnahmemodell ist für Interrupts und Traps gleichermaßen geeignet.
  - Das Beendigungsmodell ist für Interrupts und Traps gleichermaßen geeignet.
  - Bei der Behandlung einer Ausnahme nach dem Wiederaufnahmemodell wird der unterbrochene Prozess neu gestartet.
  - Das Beendigungsmodell sieht das Herunterfahren des Betriebssystems im Falle eines schwerwiegenden Fehlers vor.
- f) Welche Aussage zum Thema Adressraumschutz ist richtig? 2 Punkte
- Adressraumschutz durch Abteilung erlaubt es, mehrere Benutzerprozesse voneinander zu isolieren.
  - Adressraumschutz durch Eingrenzung benötigt keinen verschiebenden Lader, um Programmadressen an Arbeitsspeicheradressen zu binden.
  - Beim Einsatz von Segmentierung ist es möglich, dass dieselbe logische Adresse in unterschiedlichen logischen Adressräumen auf unterschiedliche physikalische Adressen verweist.
  - Die Umsetzung von logischen auf physikalische Adressen erfolgt bei segmentiertem Adressraumschutz zum Ladezeitpunkt.

g) Gegeben seien die folgenden Präprozessor-Makros:

```
#define ADD(x, y) x + y
#define SUB(x, y) x - y
```

Was ist das Ergebnis des folgenden Ausdrucks?

```
SUB(3, ADD(1, 4)) * 2
```

- 7  
 -4  
 10  
 12

2 Punkte

h) Welche Antwort trifft für die Eigenschaften eines UNIX/Linux-Dateideskriptors zu?

- Ein Dateideskriptor ist eine Verwaltungsstruktur, die auf der Festplatte gespeichert ist und Informationen über Größe, Zugriffsrechte, Änderungsdatum usw. einer Datei enthält.  
 Ein Dateideskriptor ist eine prozesslokale Ganzzahl, die der Prozess zum Zugriff auf eine geöffnete Datei benutzen kann.  
 Ein Dateideskriptor ist ein Zeiger auf Betriebssystem-interne Strukturen, der von den Systemaufrufen ausgewertet wird, um auf Dateien zuzugreifen.  
 Es ist nicht möglich ein und dieselbe Datei in mehreren Prozessen gleichzeitig zu öffnen.

2 Punkte

i) Man unterscheidet zwischen Traps und Interrupts. Welche Aussage ist richtig?

- Normale Ganzzahlarithmetik-Operationen können zu einem Interrupt führen.  
 Bei der mehrfachen Ausführung eines unveränderten Programms mit gleicher Eingabe treten Interrupts immer an den gleichen Stellen auf.  
 Traps können nicht durch Speicherzugriffe ausgelöst werden.  
 Traps stehen immer in ursächlichem Zusammenhang zu der Ausführung eines Maschinenbefehls.

2 Punkte

### Aufgabe 1.2: Mehrfachauswahl-Fragen (4 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 zu UNIX-Dateisystemen sind richtig?

- Zur Anzeige des Inhaltes einer Datei ist es notwendig, das Leserecht auf dem übergeordneten Verzeichnis zu besitzen.  
 Für das Löschen einer Datei sind die Rechte-Informationen im Dateikopf (*Inode*) der Datei irrelevant.  
 Nach dem Löschen eines Dateikopfes (*Inode*) und der dazugehörigen Datenblöcke ist es möglich, dass weiterhin *hard links* auf den Inode verweisen.  
 Ein Pfadname, der nicht mit einem '/'-Zeichen beginnt, wird relativ zum Home-Verzeichnis des Benutzers interpretiert.  
 Innerhalb eines UNIX-Dateisystembaumes können die Inhalte mehrerer Festplatten eingebunden sein.  
 In einem Namensraum mit hierarchischer Struktur ist die Verwendung von gleichen Namen in unterschiedlichen Kontexten möglich.  
 Der Name einer Datei wird getrennt von ihrem Dateikopf (*Inode*) gespeichert.  
 Auf jedes Verzeichnis verweisen immer mindestens zwei *hard-links*.

4 Punkte



**// Funktion main()**

**// Befehlszeilenargumente pruefen, weitere allgemeine  
// Vorbereitungen**

 **A:**

**// Threads starten**

**// Auf Beendigung aller Threads warten,  
// Gesamtergebnis ausgeben, aufräumen**

**// Ende Funktion main()**

 **T:**

**// Funktion runThread()**

R:

**// Ende Funktion runThread()**

**// Funktion matchEntriesInDir()**



