

Aufgabe 1.1: Einfachauswahl-Fragen (14 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 Virtuellem Speicher? 2 Punkte

- Speicher der einem Prozess durch entsprechende Hardware (MMU) und durch Ein- und Auslagern von Speicherbereichen vorgespiegelt wird, aber möglicherweise größer als der verfügbare physikalische Hauptspeicher ist.
- Speicher, der nur im Betriebssystem sichtbar ist, jedoch nicht für einen Anwendungsprozess.
- Unter einem Virtuellen Speicher versteht man einen physikalischen Adressraum, dessen Adressen durch eine MMU vor dem Zugriff auf logische Adressen umgesetzt werden.
- Virtueller Speicher kann dynamisch zur Laufzeit von einem Programm erzeugt werden (Funktion `valloc(3)`).

b) Welche Aussage über das aktuelle Arbeitsverzeichnis (Current Working Directory) trifft zu? 2 Punkte

- Jedem UNIX-Benutzer ist zu jeder Zeit ein aktuelles Verzeichnis zugeordnet.
- Besitzt ein UNIX-Prozess kein Current Working Directory, so beendet sich der Prozess mit einem Segmentation Fault.
- Mit dem Systemaufruf `chdir()` kann das aktuelle Arbeitsverzeichnis durch den Vaterprozess verändert werden.
- Pfadnamen, die nicht mit dem Zeichen `'/'` beginnen, werden relativ zu dem aktuellen Arbeitsverzeichnis interpretiert

c) In einem UNIX-UFS-Dateisystem gibt es symbolische Namen/Verweise (Symbolic Links) und feste Links (Hard Links) auf Dateien. Welche Aussage ist richtig? 2 Punkte

- Ein Symbolic Link kann nicht auf Dateien anderer Dateisysteme verweisen.
- Ein Hard Link kann nur auf Verzeichnisse nicht jedoch auf Dateien verweisen.
- Wird der letzte Symbolic Link auf eine Datei gelöscht, so wird auch die Datei selbst gelöscht.
- Für jede reguläre Datei existiert mindestens ein Hard-Link im selben Dateisystem.

d) Bei der Behandlung von Ausnahmen (Traps oder Interrupts) unterscheidet man zwei Bearbeitungsmodelle. Welche Aussage hierzu ist richtig? 2 Punkte

- Das Wiederaufnahmmodell dient zur Behandlung von Interrupts (Fortführung des Programms nach einer zufällig eingetretenen Unterbrechung). Bei einem Trap ist das Modell nicht sinnvoll anwendbar, da ein Trap deterministisch auftritt und damit eine Wiederaufnahme des Programms sofort wieder den Trap verursachen würde.
- Nach dem Beendigungsmodell werden Interrupts bearbeitet. Gibt man z. B. CTRL-C unter UNIX über die Tastatur ein, wird ein Interrupt-Signal an den gerade laufenden Prozess gesendet und dieser dadurch beendet.
- Interrupts dürfen auf keinen Fall nach dem Beendigungsmodell behandelt werden, weil überhaupt kein Zusammenhang zwischen dem unterbrochenen Prozess und dem Grund des Interrupts besteht.
- Das Betriebssystem kann Interrupts, die in ursächlichem Zusammenhang mit dem gerade laufenden Prozess stehen, nach dem Beendigungsmodell behandeln, wenn eine sinnvolle Fortführung des Prozesses nicht mehr möglich ist.

e) Welche Aussage ist in einem Monoprozessor-Betriebssystem richtig? 2 Punkte

- Ein Prozess im Zustand *blockiert* muss warten, bis der laufende Prozess den Prozessor abgibt und kann dann in den Zustand *laufend* überführt werden.
- Es befindet sich zu einem Zeitpunkt maximal ein Prozess im Zustand *laufend*.
- Ist zu einem Zeitpunkt kein Prozess im Zustand *bereit*, so ist auch kein Prozess im Zustand *laufend*.
- In den Zustand *blockiert* gelangen Prozesse nur aus dem Zustand *bereit*.

- f) Welche Antwort trifft für die Eigenschaften eines UNIX/Linux Filedeskriptors zu? 2 Punkte
- Ein Filedeskriptor ist eine prozesslokale Integerzahl, die der Prozess zum Zugriff auf eine Datei benutzen kann.
 - Filedeskriptoren sind Zeiger auf Betriebssystem-interne Strukturen, die von den Systemaufrufen ausgewertet werden, um auf Dateien zuzugreifen.
 - Ein Filedeskriptor ist eine Integerzahl, die über gemeinsamen Speicher an einen anderen Prozess übergeben werden kann, und von letzterem zum Zugriff auf eine geöffnete Datei verwendet werden kann.
 - Beim Öffnen ein und derselben Datei erhält ein Prozess jeweils die gleiche Integerzahl als Filedeskriptor zum Zugriff zurück.
- g) 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).
 - Wenn ein Programm nur einen aktiven Ablauf enthält, nennt man diesen Prozess, enthält das Programm mehrere Abläufe, nennt man diese Threads.
 - Ein Prozess ist ein Programm in Ausführung - ein Prozess kann aber auch mehrere verschiedene Programme ausführen
 - Ein Programm kann immer nur von einem Prozess ausgeführt werden

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 zum Thema Threads sind richtig? 4 Punkte
- Bei User-Threads ist die Schedulingstrategie keine Funktion des Betriebssystemkerns.
 - Kern-Threads blockieren sich bei blockierenden Systemaufrufen gegenseitig.
 - Kern-Threads können Multiprozessoren nicht ausnutzen.
 - Zur Umschaltung von User-Threads ist ein Adressraumwechsel erforderlich.
 - Die Umschaltung von User-Threads ist sehr effizient.
 - Zu jedem Kern-Thread gehört ein eigener Adressraum.
 - Bei Kern-Threads ist die Schedulingstrategie meist durch das Betriebssystem vorgegeben.
 - Die Umschaltung von Kern-Threads muss immer im Systemkern erfolgen.

/ Argumente pruefen und weitere Initialisierungen */*

.....
.....
.....
.....
.....

.....
.....
.....
.....

/ Quellverzeichnis durchsuchen, Eintraege verarbeiten */*

.....
.....
.....
.....

.....
.....
.....
.....
.....
.....
.....
.....
.....

/ Auf Terminierung der Threads warten, Ergebnis ausgeben */*

.....
.....
.....
.....

/ Aufräumen */*

.....
.....
.....
.....

/ Ende Funktion main */*

Aufgabe 3: (19 Punkte)

a) Gegeben sei das nachfolgende Programm.

Skizzieren Sie den Aufbau des logischen Adressraums eines Prozesses, der dieses Programm ausführt.

- Wozu werden welche Segmente angelegt?

.....

.....

.....

.....

.....

- Zeichnen Sie für jede Variable ein, wo diese ungefähr im logischen Adressraum zu finden sein wird.

- Wohin zeigen die Zeiger m, p und q (ungefähre Position einzeichnen!)?

```
int ga;
static int gb;
char *m = "message";

int main() {
    int la;
    static int lb;
    int *p;
    static double *q;

    p = malloc(100 * sizeof(int));
    q = malloc(100 * sizeof(double));
}
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

b) Welche Arten von Schutz gewährleisten logische Adressräume?

.....

.....

.....

.....

.....

c) Wenn das folgende Programmstück in einem UNIX-System abläuft, wird ein Fehler auftreten.

```
...
int *p = NULL;
...
*p = -1;
...
```

Bitte möglichst knappe, aber präzise Antworten:

c1) Welcher Fehler tritt auf?

.....

.....

c2) Warum tritt der Fehler auf?

.....

.....

c3) Der Fehler wird von einer Hardwarekomponente zuerst entdeckt - welche Komponente ist das?

.....

.....

c4) Mit welchem Mechanismus wird der Fehler dem Betriebssystem mitgeteilt?

.....

.....

c5) Was macht das Betriebssystem mit dem Prozess, der gerade das Programmstück ausführt?

.....

.....

Aufgabe 4: (6 Punkte)

Wie wird eine Ausnahmebehandlung (Trap oder Interrupt) von der CPU und vom Betriebssystem abgewickelt?

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....