

Aufgabe 1: Ankreuzfragen (7 Punkte)

1) Einfachauswahlfragen (4 Punkte)

Bei den Einfachauswahlfragen in dieser Aufgabe ist jeweils nur **eine** richtige Antwort eindeutig anzukreuzen. Auf die richtige Antwort gibt es die angegebene Punktzahl.

Wollen Sie eine 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 Aussage ist in einem Monoprocessor-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.
- In den Zustand *blockiert* gelangen Prozesse nur aus dem Zustand *bereit*.
- Ist zu einem Zeitpunkt kein Prozess im Zustand *bereit*, so ist auch kein Prozess im Zustand *laufend*.
- Es befindet sich zu jedem Zeitpunkt maximal ein Prozess im Zustand *laufend*.

b) Namensräume dienen u. a. der Organisation von Dateisystemen. Welche Aussage ist richtig?

2 Punkte

- In einem hierarchisch organisierten Namensraum dürfen gleiche Namen in unterschiedlichen Kontexten enthalten sein.
- Der Nachteil von hierarchischen Namensräumen besteht darin, dass das Dateisystem spezielle Funktionen zum Auflösen von Namenskonflikten implementieren muss.
- Hierarchische Namensräume werden erzeugt, indem man in einem Kontext symbolische Verweise auf Dateien einträgt.
- Flache Namensräume sind besonders einfach implementierbar und damit vor allem für Mehrbenutzersysteme gut geeignet.

2) Mehrfachauswahlfragen (3 Punkte)

Bei den Mehrfachauswahlfragen in dieser Aufgabe sind jeweils m Aussagen angegeben, davon sind n ($0 \leq n \leq m$) Aussagen richtig. Kreuzen Sie alle richtigen Aussagen an.

Jede korrekte Antwort in einer Teilaufgabe gibt einen Punkt, jede falsche Antwort einen 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) Gegeben sei folgendes Programmfragment:

3 Punkte

```
static int a = 2018;

int f1 (int const *y) {
    int c;
    char *d = malloc(1337);
    int (*e)(const int *) = f1;
    y++;
    // ...
}
```

Welche der folgenden Aussagen zum obigen Programmfragment sind richtig?

- a liegt im Datensegment.
- c ist mit dem Wert 0 initialisiert.
- Die Speicherstelle, auf die d zeigt, verliert beim Rücksprung aus der Funktion f1() ihre Gültigkeit.
- e liegt im Stacksegment und zeigt in das Textsegment.
- Die Anweisung y++ führt zu einem Laufzeitfehler, da y konstant ist.
- d ist ein Zeiger, der in den Heap zeigt.

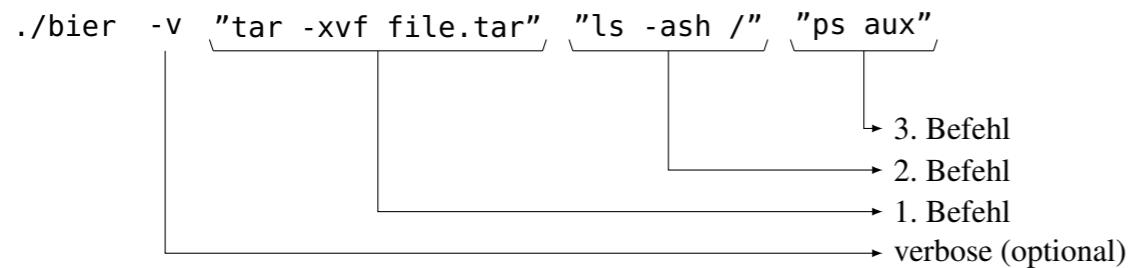
Aufgabe 2: bier (16 Punkte)

Sie dürfen diese Seite zur besseren Übersicht bei der Programmierung heraustrennen!

Schreiben Sie ein Programm `bier` (**B**inary **E**xecutable **R**unner), das alle per Befehlszeilenargument übergebenen Befehle der Reihe nach ausführt. Zusätzlich kann der Verbose-Modus mittels des Arguments `-v` aktiviert werden. Zur Vereinfachung dürfen Sie davon ausgehen, dass `-v` immer als erstes Argument übergeben wird. Im Verbose-Modus soll `bier` am Ende der Ausführung des jeweiligen Befehls die folgenden Informationen auf `stderr` schreiben:

- die komplette Befehlszeile
- die Prozess-ID (PID)
- den Exitstatus

Beispielhafter Aufruf von `bier` mit 3 auszuführenden Befehlen:



Das Programm soll wie folgt aufgebaut sein:

- Die Hauptfunktion (`main`) prüft ob mindestens ein Befehl übergeben wurde, wertet das optionale `-v` aus und ruft für jede übergebene Befehlszeile die Funktion `rum` auf. Falls kein Befehl übergeben wird, soll die Ausführung mittels Aufruf von `ouzo()` abgebrochen werden.
- Funktion `void rum(const char *cmdline)`: Führt die übergebene Befehlszeile in einem neuen Prozess aus. Dazu werden das auszuführende Programm und die Parameter mittels `strtok(3)` aus der Befehlszeile extrahiert und mithilfe einer Funktion der `exec()`-Familie ausgeführt. Zur Vereinfachung dürfen Sie annehmen, dass alle zu extrahierenden Parameter einer Befehlszeile durch Leerzeichen voneinander getrennt sind und sich innerhalb der einzelnen Parameter keine weiteren Leerzeichen befinden.

Achten Sie auf korrekte Ressourcenverwaltung.

Hinweise:

- Zur Vereinfachung dürfen Sie die Ausführung im Fehlerfall mittels `die()` abbrechen.
- Hinweise zur Benutzung von `bier` können durch Aufruf von `ouzo()` ausgegeben werden.

```

#include <stdlib.h>
#include <stdio.h>
#include <stdbool.h>
#include <limits.h>
#include <errno.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <sys/wait.h>

static void die(const char *msg) {
    perror(msg); exit(EXIT_FAILURE);
}

static void ouzo(void) {
    fprintf(stderr, "Usage: _bier_ [-v]_<commands>\n");
    exit(EXIT_FAILURE);
}

// Funktions- & Strukturdekl., globale Variablen, etc.
-----
-----
-----
// Hauptfunktion (main)
int main(int argc, char *argv[]) {
-----
-----
-----
-----
-----
-----
-----
-----
-----
return EXIT_SUCCESS;
}

```

M:

// Funktion rum

Aufgabe 3: Prozesszustände (7 Punkte)

Beschreiben Sie die Prozesszustände bei der Einplanung von Prozessen sowie die Ereignisse, die jeweils zu Zustandsübergängen führen (Skizze mit kurzer Erläuterung der Zustände und Übergänge).

