

Friedrich-Alexander-Universität Erlangen-Nürnberg

**Miniklausur zum Modul
„Systemprogrammierung“**

	erreichbare Punkte	erhaltene Punkte							
Aufgabe 1	6								
Aufgabe 2	15								
Aufgabe 3	9	a	b	c					
Summe	30								
Note									

_____ (Name)

_____ (Vorname)

_____ (Matrikel-Nr.)

_____ (Studiengang)

_____ (Semester)

Durch meine Unterschrift bestätige ich

- den Empfang der vollständigen Klausurunterlagen (7 Seiten inklusive Deckblatt),
- den Empfang der Manualseiten (1 Blatt mit 3 Manualseiten: opendir/readdir, printf, stat)
- die Kenntnisnahme der Hinweise auf Seite 2.

Erlangen, 10.05.2017

(Unterschrift)

Hinweise

Bitte lesen Sie die folgenden Informationen aufmerksam und unterschreiben Sie die Erklärung auf der ersten Seite.

- Die Bearbeitungszeit beträgt 30 Minuten.
- Es sind **keine** eigenen Hilfsmittel zugelassen mit Ausnahme eines einzelnen, handbeschriebenen Notizblatts.
- Die Lösung einer Aufgabe soll auf das Aufgabenblatt in den dafür vorgesehenen Raum geschrieben werden. Beachten Sie bitte, dass der freigelassene Platz großzügig bemessen ist und nicht unbedingt der erwarteten Antwortlänge entspricht. Sollte der Platz nicht ausreichen, können Sie die Rückseiten der Aufgabenblätter mitverwenden. Kennzeichnen Sie dabei die Zugehörigkeit Ihrer Lösung zu einer Aufgabe. Bei Bedarf können zusätzliche Lösungsblätter (weiß) ausgeteilt werden. Vermerken Sie vor deren Verwendung unbedingt Ihren Namen und Ihre Matrikelnummer darauf!
- Schmierpapier und Manuseiten dürfen **nicht** abgegeben werden. Bei Bedarf ist von der Aufsicht weiteres Schmierpapier (farbig) erhältlich.
- Fragen zu den Prüfungsaufgaben können grundsätzlich nicht beantwortet werden.
- Tragen Sie Ihren Namen und Vornamen, Ihre Matrikelnummer, Studiengang und Fachsemesterzahl auf dem Deckblatt der Klausur ein.
- Bitte legen Sie Ihren Studenten- und einen Lichtbildausweis zur Kontrolle bereit.
- Sie dürfen den Raum nicht verlassen, bevor Ihre Personalien überprüft wurden und Sie die Klausurunterlagen der Aufsicht zurückgegeben haben.
- In den letzten 15 Minuten der Bearbeitungszeit können Sie den Raum nicht mehr verlassen. Bleiben Sie an Ihrem Platz sitzen, bis schließlich alle Klausurunterlagen eingesammelt sind und die Aufsicht das Zeichen zum Gehen gibt.
- Die **Ergebnisse der Miniklausur** werden in ca. einer Woche analog zu den Bewertungen der Übungsaufgaben im WAFFEL zugänglich gemacht:
<https://waffel.informatik.uni-erlangen.de>

Aufgabe 1.1: Einfachauswahl-Fragen (3 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 Aussage zu Zeigern in C-Programmen ist richtig?

1,5 P.

- Ein Zeiger kann zur Manipulation von Daten in schreibgeschützten Speicherbereichen verwendet werden.
- Zeiger vom Typ `void *` existieren in C nicht, da solche Zeiger auf "Nichts" keinen sinnvollen Einsatzzweck hätten.
- Die Übergabesemantik für Zeiger als Funktionsparameter ist *call-by-reference*.
- Zeiger können verwendet werden, um in C eine *call-by-reference*-Übergabesemantik nachzubilden.

b) Welche Aussage zum Thema Betriebsarten ist richtig?

1,5 P.

- Mehrprogrammbetrieb ermöglicht die Ausführung mehrerer Programme innerhalb des selben Prozesses.
- Echtzeitsysteme findet man hauptsächlich auf großen Serversystemen, die eine enorme Menge an Anfragen zu bearbeiten haben.
- Mehrzugangsbetrieb ist nur in Verbindung mit CPU- und Speicherschutzmechanismen sinnvoll realisierbar.
- Beim Stapelbetrieb können keine globalen Variablen existieren, da alle Daten im Stapelsegment (Stack) angelegt sind.

Aufgabe 1.2: Mehrfachauswahl-Fragen (3 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 Dateisysteme sind richtig? 3 Punkte
- Wird der letzte *hard-link* auf eine Datei entfernt, so wird auch die Datei selbst gelöscht.
 - In einem hierarchisch organisierten Dateisystem dürfen gleiche Dateinamen in unterschiedlichen Verzeichnissen enthalten sein.
 - Zum Anlegen oder Löschen von Dateien sind die Schreibzugriffsrechte auf das übergeordnete Verzeichnis irrelevant.
 - Ein *symbolic link* kann auf eine Datei innerhalb eines anderen Dateisystems verweisen.
 - Wird die Datei gelöscht, auf die ein *symbolic link* verweist, so wird auch der *symbolic link* selbst gelöscht.
 - Auf das Wurzelverzeichnis (*root directory*, "/") darf immer nur genau ein *hard-link* verweisen.

Aufgabe 2: (15 Punkte)

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

Schreiben Sie ein Programm `dirlist`, welches - analog zum aus Linux bekannten Befehlszeilentool `ls` - den Inhalt eines Verzeichnisses auflistet, das dem Programm als einziger Kommandozeilenparameter übergeben wird. Sie dürfen zur Vereinfachung davon ausgehen, dass das Programm immer mit einem Parameter aufgerufen wird.

Zusätzlich zum Dateinamen sollen auch der Typ sowie die Größe (in Bytes) des jeweiligen Eintrags, und zuletzt die Summe der Größen aller untersuchten Einträge auf der Standardausgabe ausgegeben werden. Folgende Typen sind bekannt:

- 'D': Verzeichnis
- 'R': reguläre Datei
- 'L': symbolischer Link
- '?': keiner der obigen Typen

Beispielausgabe eines Aufrufs (`ls dirname`):

```
file.txt R 123B
symlink.csv L 8B
. D 4096B
.. D 4096B
Total size: 8323B
```

Achten Sie auf korrekte Fehlerbehandlung!

```
#include <dirent.h>
#include <errno.h>
#include <unistd.h>
#include <limits.h>
#include <stdbool.h>
#include <stdio.h>
#include <stdlib.h>
#include <sys/stat.h>
#include <sys/types.h>

static void die(const char msg[]) {
    perror(msg);
    exit(EXIT_FAILURE);
}
```


Aufgabe 3: (9 Punkte)

a) In der Systemprogrammierung werden zwei Kategorien von Programmunterbrechungen (Ausnahmen) unterschieden. Welche sind dies? Grenzen Sie diese kurz voneinander ab. (2 Punkte)

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

b) Wenn das folgende Programmstück in einem UNIX-System abläuft, wird ein Fehler auftreten: (4 Punkte)

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

b1) Warum tritt bei der Ausführung des Programmstückes ein Fehler auf?

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

b2) Welche Hardwarekomponente entdeckt diesen Fehler zuerst und wie teilt sie diesen Fehler dem Betriebssystem mit?

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

b3) Was macht das Betriebssystem mit dem Prozess, der gerade das Programmstück ausführt?

.....
.....

c) Zur Behandlung von Unterbrechungen werden Mantelprozeduren (*wrapper*) angesprungen. Welche Arbeitsschritte sind in diesen Prozeduren notwendig, um die Unterbrechung korrekt zu behandeln? (3 Punkte)

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