

**Aufgabe 1: Ankreuzfragen (22 Punkte)**

## 1) Einfachauswahlfragen (18 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) Wie funktioniert Adressraumschutz durch Eingrenzung?

2 Punkte

- Der Lader positioniert Programme immer so im Arbeitsspeicher, dass unerlaubte Adressen mit nicht-existierenden physikalischen Speicherbereichen zusammenfallen.
- Die MMU kennt die Länge eines Segments und verhindert Speicherzugriffe darüber hinaus.
- Begrenzungsregister legen einen Adressbereich im physikalischen Adressraum fest, auf den alle Speicherzugriffe beschränkt werden.
- Jedes Programm bekommt zur Ladezeit mehrere Wertepaare aus Basis- und Längenregistern zugeordnet, die die Größe aller Segmente des darin laufenden Prozesses festlegen.

b) Welche der folgenden Aussagen trifft auf das Programmfragment zu?

2 Punkte

```
int f1 (const int *y) {
    static int b;
    char *d = malloc(2407);
    int (*e)(const int *) = f1;
    b += *y;
    y++;
    return b;
}
```

- Die Variable b liegt im Stacksegment.
- Die Anweisung y++ führt zu einem Laufzeitfehler, da y konstant ist.
- Die Speicherstelle, auf die d zeigt, verliert beim Rücksprung aus der Funktion f1() ihre Gültigkeit.
- Die Variable e liegt im Stacksegment und zeigt auf eine Stelle im Textsegment.

c) Welcher UNIX-Systemaufruf wird bei der Verwendung von Sockets auf keinen Fall gebraucht?

2 Punkte

- shutdown()
- wait()
- close()
- accept()

h) Welche Aussage zum Thema Betriebsarten ist richtig?

2 Punkte

- Echtzeitsysteme findet man hauptsächlich auf großen Serversystemen, die eine enorme Menge an Anfragen zu bearbeiten haben.
- Beim Stapelbetrieb können keine globalen Variablen existieren, weil alle Daten im Stapel-Segment (Stack) abgelegt sind.
- Mehrprogrammbetrieb ermöglicht die simultane Ausführung mehrerer Programme innerhalb desselben Prozesses.
- Mehrzugangsbetrieb ist nur in Verbindung mit CPU- und Speicherschutz sinnvoll realisierbar.

i) Wie viele Prozesse existieren nach erfolgreicher Ausführung der folgenden Programmsequenz?

2 Punkte

```
if (fork())  
    fork();  
exit(EXIT_SUCCESS);
```

- 0
- 1
- 2
- 3

**Aufgabe 2: wosch (45 Punkte)**

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

Um den immer häufiger werdenden Anglizismen in der deutschen Sprache entgegenzuwirken, ist es Ihre Aufgabe, den **Wortverschönerer**, kurz **wosch**, zu implementieren. Dieser ersetzt mithilfe eines Glossars Begriffe durch eine entsprechende Verschönerung.

Ihr Programm nimmt als ersten Parameter den Pfad zum Glossar entgegen, welches die Verschönerungen von Begriffen enthält. Alle weiteren Parameter werden als zu bearbeitende Pfade interpretiert. Falls keine weiteren Parameter neben dem Pfad zum Glossar vorhanden sind, soll die Standardeingabe verarbeitet werden. Die Ausgabe der verschönerten Texte geschieht auf der Standardausgabe.

Beispiel: `./wosch glossar.wosch text.txt`

glossar.wosch:	text.txt:	Ausgabe:
Hello Hallo	Hello World,	Hallo Welt,
World Welt	wie geht es?	wie geht es?
Nice Nette	Nice Aufgabe, oder?	Nette Aufgabe, oder?

**Implementieren Sie zur Umsetzung folgende Funktionen:**

**int main(int argc, char \*argv[])**

Liest mit der Funktion `glossar` das Glossar ein. Danach werden die Verschönerungen mithilfe von `lese_ein` und `bearbeite` vorgenommen. Abschließend werden noch belegte Ressourcen freigegeben.

**struct glossar \*glossar(const char \*p)**

Speichert alle vorkommenden Verschönerungen aus der Datei hinter dem Pfad `p` in einem **struct** `glossar` und gibt einen Zeiger darauf zurück. Die Einträge im **struct** `glossar` sollen mithilfe der gegebenen Vergleichsfunktion `vergleiche` sortiert werden. Eine valide Zeile besteht aus einem zu ersetzenden Begriff (siehe Hinweis 1) und der zugehörigen, beliebig langen Verschönerung. Beide Bestandteile sind mit Leerzeichen und/oder Tabulatoren voneinander getrennt. Jede Zeile der Glossardatei beinhaltet maximal 200 Zeichen, ebenso kommt kein zu ersetzender Begriff doppelt vor. Nicht valide Zeilen werden ignoriert.

**void bearbeite(struct glossar \*g, const char \*p)**

Bearbeitet einen übergebenen Pfad `p` mit dem übergebenen **struct** `glossar *g`: Bei einer regulären Datei wird `lese_ein` aufgerufen, allerdings nur, sofern diese auf `.txt` endet. Verzeichnisse werden rekursiv abgearbeitet. Alle anderen Dateien werden ignoriert, insbesondere auch symbolische Verknüpfungen.

**void lese\_ein(struct glossar \*g, FILE \*in, FILE \*out)**

Liest den Inhalt vom übergebenen `FILE *in` ein, und gibt diesen auf `FILE *out` aus. Dabei werden Verschönerungen von Begriffen mithilfe von `verschoenere` realisiert.

**const char \*verschoenere(struct glossar \*g, const char \*str)**

Überprüft, ob der Begriff `c` im übergebenen `glossar` vorhanden ist. Wenn ja, wird dieser zurückgegeben, anderenfalls der Begriff selbst.

**Hinweise:**

1. Begriffe bestehen stets aus nicht mehr als 50 Zeichen, welche nicht Leerzeichen, Tabulatoren oder Zeilenumbrüche sind.
2. Sollte der Bearbeitungsprozess bei einer Datei fehlschlagen, so soll dies zwar mit einer Fehlermeldung gemeldet werden, jedoch nicht die Bearbeitung anderer Dateien weiter beeinflussen.

```
int main(int argc, char *argv[]) {
```

```
}
```

<b>M:</b>
-----------

```
static struct glossar *glossar(const char *p) {
```

```
    // Glossar anlegen
```

```
    // Datei öffnen
```





2) Beschreiben Sie was geschieht, wenn ein Prozess auf eine ausgelagerte Seite zugreift. (3 Punkte)

-----  
-----  
-----  
-----

3) Nehmen Sie an, dass das folgende Programmstück in einem UNIX-System abläuft: (3 Punkte)

```
// ...  
int *p = malloc(sizeof(int));  
*p = -1;  
// ...
```

a) Welcher Fehler kann dabei auftreten? (1 Punkt)

-----  
-----

b) Der Fehler wird von einer Hardwarekomponente zuerst entdeckt - welche Komponente ist das? (1 Punkt)

-----  
-----

c) Was wird das Betriebssystem mit dem Prozess machen, der den Fehler hervorruft? (1 Punkt)

-----  
-----