Dr. Andreas Borchert Institut für Angewandte Informationsverarbeitung 9. November 2009
Wolfgang Kaifler / Michael Mattes Blatt 5

Systemnahe Software I
im Wintersemester 2009/2010

Abgabetermin: bis 17. November 2009

12 Stringfunktionen (6 Punkte)

Schreiben Sie ein Programm `strfun.c`, in welchem Sie die Funktionsweise und Rückgabewerte der String-Funktionen `strlen()`, `strcmp()` und `strcpy()` auswerten. Eine Beispielausgabe sehen Sie bei der nächsten Aufgabe.

13 Eigene Implementierung (12 Punkte)

Implementieren Sie danach eigene Varianten der drei Funktionen, die Sie mit dem Präfix `my_` kennzeichnen. So soll Ihre Variante von `strlen()` einfach `my_strlen()` heissen. Ausser dem Namen sollen Sie die Signatur der Funktionen unverändert übernehmen.

Die Ausgabe Ihres Programms sollte etwa so aussehen:

```
$ ./strfun
strlen("Hallo Welt!") = 11
strcpy(buffer1, "Hallo ") = "Hallo "
strcpy(buffer2, "Welt!") = "Welt!"
strcmp(buffer1, buffer2) = -15
strcmp(buffer1, "Hallo ") = 0
my_strlen("Hallo Welt!") = 11
my_strcpy(buffer1, "Hurra ") = "Hurra "
my_strcpy(buffer2, "Wirt!") = "Wirt!"
my_strcmp(buffer1, buffer2) = -15
my_strcmp(buffer1, "Hurra ") = 0
```

Hinweis: Die Signaturen können Sie den Manpages der Funktionen entnehmen (z.B. `man -s3c strcpy` bzw. `man 3 strcpy`).

14 Geldanlagensucher (24 Punkte)

Schreiben Sie ein Programm `geld.c`, welches verschiedene Geldanlagen einliest und dann anhand der Benutzereingaben die beste verfügbare Geldanlage (inklusive der Geldsumme am Ende der Laufzeit) ausgibt.

Verwenden Sie für die Verwaltung der Geldanlagen ein Array von `geldanlage`-Verbünden. Das Array darf keine globale Variable sein! Die Definition des Arrays ist wie folgt vorgegeben:

```
#define MAXANLAGEN (100)

typedef struct _geldanlage {
        int nummer; // Eine frei waehlbare Nummer
        float anfangszinssatz; // Zinssatz (in % p.a.) der ersten zwei Jahre
        float restzinssatz; // Zinssatz danach
        int mindesteinlage; // Mindesteinlage in Zehntel-Cents
} geldanlage;


int main() {
        geldanlage anlagen[MAXANLAGEN];
        // weiter
}
```

Die vom Benutzer abzufragenden Eingaben bestehen aus der Anlagesumme in Euro und der Laufzeit in Jahren. Lagern Sie das Einlesen der Geldanlagen sowie der Benutzereingaben in eigene Funktionen aus. Die Signatur der Anlagen-Einlesefunktion könnte wie folgt aussehen:

```
// Liest bis zu maxAnlagen Geldanlagen von der Standardeingabe
// und liefert die Anzahl der Anlagen zurueck
int anlagenEinlesen(geldanlage *anlagen, int maxAnlagen);
```

Hinweis: Rechnen Sie intern in Zehntel-Cent, aber geben Sie die Beträge centgenau und abgerundet aus.

Beispiele:

```
$ ./geld
Freie Nummer der Anlage (-1 beendet Einlesen): 1
Zinssatz der ersten zwei Jahre in % p.a.: 5.2
Zinssatz danach: 1.8
Mindesteinlage in Euro: 500
Freie Nummer der Anlage (-1 beendet Einlesen): -1
Habe 1 Geldanlagen eingelesen.
Anlagesumme in Euro: 500
Laufzeit in Jahren: 3

Fuer eine Summe von 500.00 Euro ist Anlage 1 die beste.
Nach 3 Jahren betraegt das Guthaben 563.31 Euro.
```

```
$ ./geld
Freie Nummer der Anlage (-1 beendet Einlesen): -1
Habe 0 Geldanlagen eingelesen.
Anlagesumme in Euro: 100
Laufzeit in Jahren: 1

Es wurde keine passende Anlage gefunden!

$ ./geld
Freie Nummer der Anlage (-1 beendet Einlesen): 42
Zinssatz der ersten zwei Jahre in % p.a.: 2
Zinssatz danach: 2
Mindesteinlage in Euro: 1000
Freie Nummer der Anlage (-1 beendet Einlesen): -1
Habe 1 Geldanlagen eingelesen.
Anlagesumme in Euro: 999
Laufzeit in Jahren: 10

Es wurde keine passende Anlage gefunden!

# Mit Hilfe der Beispieldatei anlagen.txt
$ ./geld <anlagen.txt
[... Eingabeprompts ...]
Fuer eine Summe von 3200.00 Euro ist Anlage 3 die beste.
Nach 6 Jahren betraegt das Guthaben 3711.01 Euro.

# Nachdem die Laufzeit in der Datei auf 4 Jahre reduziert wurde:
$ ./geld <anlagen.txt
[... Eingabeprompts ...]
Fuer eine Summe von 3200.00 Euro ist Anlage 2 die beste.
Nach 4 Jahren betraegt das Guthaben 3561.67 Euro.
```

Viel Erfolg!