Fakultät IV Elektrotechnik und Informatik
Einführung in die Programmierung WS 2023/2024
Manfred Hauswirth
Damien Foucard / Uwe Kuehn / Aljoscha Meyer

Programmierblatt 03

Ausgabe: *23.11.2023 16:00*
Abgabe: *08.12.2023 20:00*

Thema: *Listen, Datenstrukturen*

Abgabemodalitäten

1. Alle abzugebenden Quelltexte müssen ohne Warnungen und Fehler auf Ihrem Rechner mit dem Befehl `clang -std=c11 -Wall -g` kompilieren.
2. Die Abgabe für den Quellcode erfolgt ausschließlich über unser Git im entsprechenden Branch. Nur wenn ein Ergebnis im ISIS-Kurs angezeigt wird, ist sichergestellt, dass die Abgabe erfolgt ist. Die Abgabe ist bestanden, wenn Sie an Ihrem Test einen grünen Haken sehen.
3. Sie können bis zur Abgabefrist beliebig oft neue Versionen abgeben. Lesen Sie sich die Hinweise der Tests genau durch, denn diese helfen Ihnen die Abgabe zu korrigieren.
   Bitte beachten Sie, dass ausschließlich die letzte Abgabe gewertet wird.
4. Die Abgabe erfolgt, sofern nicht anders angegeben, in folgendem Branch: `iprg-b<xx>-a<yy>`, wobei `<xx>` durch die zweistellige Nummer des Aufgabenblattes und `<yy>` durch die entsprechende Nummer der Aufgabe zu ersetzen sind.
5. Geben Sie für jede Aufgabe die Quellcodedatei(en) gemäß der Vorgabe ab. Im ISIS-Kurs werden zum Teil Vorgabedateien bereitgestellt. Nutzen Sie diese zur Lösung der Aufgaben.
6. Die Abgabefristen werden vom Server überwacht. Versuchen Sie Ihre Abgabe so früh wie möglich zu bearbeiten. Damit minimieren Sie auch das Risiko, die Abgabefrist auf Grund von „technischen Schwierigkeiten“ zu versäumen. Eine Programmieraufgabe gilt als bestanden, wenn alle bewerteten Teilaufgaben bestanden sind.
7. Sofern die Aufgabenstellenstellung nichts gegenteiliges besagt, dürfen keine weiteren `include` Direktiven verwendet werden, d.h., es dürfen keine zusätzlichen Bibliotheksfunktionen verwendet werden. Eigene Funktionen zu implementieren und verwenden ist hingegen legitim und häufig eine gute Idee für besser lesbaren Code.

Aufgabe 1 Eine Büchersammlung als Liste (bewertet)

Ziel der Aufgabe ist es, Daten zu Büchern aus einer Datei zu laden, diese in einer einfach verkettete Liste zu verwalten und diese mithilfe einer vorgegebenen Funktion auszugeben.

Bei den Daten handelt es sich um den Titel, den Autor, das Erscheinungsjahr und die ISBN des Buches. Diese Daten werden dabei aus der Datei `buecherliste.txt` eingelesen.

Listing 1: buecherliste.txt

```
1984;George Orwell;1949;9783548267456;
A Scanner Darkly;Phillip K. Dick;1973;9780547572178;
Bodyguard;William C. Dietz;1994;9780441001057;
```

Beachten Sie: Es müssen die folgende Datenstruktur und Funktionen implementiert werden:

- `struct _element` ist in der Header-Datei zu implementieren
- Die übrigen Funktionen Ihrer Bibliothek schreiben Sie in der C-Datei:
- `element* insert_at_begin(element* , element* )`
- `element* construct_element(char* , char* , uint32_t , uint64_t )`
- `void free_list(list* )`
- Wählen Sie geeignete „sprechende“ Variablennamen und achten Sie auf die korrekte Reihenfolge (Semantik) der Funktionsparameter. Die Vorlage kann aktuell nicht kompiliert werden.

Dagegen dürfen die `main()` Funktion und die übrigen Funktionen **nicht** geändert werden. Auch hier wird eine Bibliothek eingebunden (`introprog_structs_lists_input.c` und `introprog_structs_lists_input.h`), wie aus dem folgenden beispielhaften Programmaufruf ersichtlich wird:

Listing 2: Programmbeispiel

```
1 >clang -std=c11 -Wall buecherliste.c introprog_buecherliste.c \
2     introprog_structs_lists_input.c -o introprog_buecherliste
3 >./introprog_buecherliste buecherliste.txt
4 Meine Bibliothek
```

```
================

Buch 1
    Titel: Mars Plus
    Autor: Frederick Pohl
    Jahr:  1994
    ISBN:  9780671876050
Buch 2
    Titel: Man Plus
    Autor: Frederick Pohl
    Jahr:  1976
    ISBN:  9780765321787
[etc.]
```

Wichtig: Die Aufgabe muss den folgenden Anforderungen genügen:

- Das `struct _element` muss die folgenden Variablen beinhalten:
  - Ein `char` Array `title`, statisch für die Größe MAX_STR reserviert.
  - Ein `char` Array `author`, statisch für die Größe MAX_STR reserviert.
  - Eine `uint32_t` Variable `year`.
  - Eine `uint64_t` Variable `isbn`.
  - Ein Pointer `next` auf das nächste Element.
- Neue Elemente sollen stets an den Anfang der Liste platziert werden, sodass das neue Element jeweils die Stelle von `first` einnimmt.
- Der bestehende Code, außerhalb der geforderten Änderungen, darf nicht verändert werden. Insbesondere dürfen die Funktionen `construct_list`, `read_list` und `main` und die Datenstruktur `struct _list` (inkl. dem `typedef`) nicht verändert werden.
- Der Speicher dynamisch reservierte Speicher soll ohne Ausnahme (auch `list *alist`) freigegeben werden.
- Im Ordner der Vorgaben liegen noch zwei weitere beispielhafte Eingabedateien, nämlich `buecherliste.evil.txt` und `buecherliste.random12342.txt`. Diese enthalten größere Beispiele. In der `buecherliste.evil.txt` ist ein Buch enthalten, welches einen sehr langen Namen (ca. 900 Zeichen) hat. Beachten Sie, dass wir erwarten, dass dieser Name abgeschnitten wird und bei der Ausgabe nur die ersten 254 Zeichen ausgegegeben werden. Testen Sie ihr Programm auch mit diesen Eingabedateien.
- Überprüfen Sie mit `valgrind`, ob korrekt auf den Speicher zugegriffen wird.

Nutzen Sie zur Lösung der Aufgabe die Vorgaben aus unserem ISIS-Kurs. Fügen Sie Ihre Lösung als Datei `introprog_buecherliste.c` und `introprog_buecherliste.h` im entsprechenden Abgabebranch in Ihrem persönliches Repository ein und übertragen Sie die Lösung an die Abgabeplattform.

Aufgabe 2 Eine Büchersammlung als sortierte Liste (bewertet)

Die Elemente sollen nun aufsteigend sortiert nach ISBN in die einfach verkettete Liste eingetragen werden. Übernehmen Sie die in der letzten Aufgabe entwickelte Bibliothek. Ergänzen Sie nun folgende Funktion zusätzlich in einer zweiten Bibliothek:

- `element* insert_sorted(element*, element *)`

Der Rest des Codes soll nicht angepasst werden. Auch hier wird die Bibliothek eingebunden (`introprog_structs_lists_input.c` und `introprog_structs_lists_input.h`), wie aus dem folgenden beispielhaften Programmaufruf ersichtlich wird:

Listing 3: Programmbeispiel

```
> clang -std=c11 -Wall buecherliste.c introprog_buecherliste.c \
      introprog_sortierte_buecherliste.c \
      introprog_structs_lists_input.c \
      -o introprog_sortierte_buecherliste
> ./introprog_sortierte_buecherliste buecherliste.txt
Meine Bibliothek
================

Buch 1
    Titel: Neuromancer
    Autor: William Gibson
    Jahr:  1984
    ISBN:  9780006480419
Buch 2
    Titel: Burning Chrome
    Autor: William Gibson
    Jahr:  1986
    ISBN:  9780060539825
Buch 3
    Titel: Interface
    Autor: Neal Stephenson
    Jahr:  1994
    ISBN:  9780099427759
[etc.]
```

Wichtig: Die Aufgabe muss den folgenden Anforderungen genügen:

- Neue Elemente sollen so in die Liste eingefügt werden, dass die Elemente aufsteigend nach ISBN sortiert sind. (D.h. zu jedem Zeitpunkt gilt das erste Buch hat die kleinste ISBN und jedes darauffolgende Buch hat eine größer werdende ISBN.)
- Abgesehen von der Ordnung der Elemente gelten alle Anforderungen aus der vorherigen Aufgabe.

Nutzen Sie zur Lösung der Aufgabe die Vorgaben aus unserem ISIS-Kurs. Fügen Sie Ihre Lösung als Datei `introprog_buecherliste.c`, `introprog_buecherliste.h`, `introprog_sortierte_buecherliste.c` und `introprog_sortierte_buecherliste.h` im entsprechenden Abgabebranch in Ihrem persönliches Repository ein und übertragen Sie die Lösung an die Abgabeplattform.