

Programmmentwurf Systemnahe Programmierung Kurs TIT20

Bearbeitungshinweise

Die Prüfungsleistung für die Vorlesung Systemnahe Programmierung wird durch einen Programmmentwurf in Intel x86-64 Assembler für den NASM Assembler unter dem Betriebssystem Linux erbracht. Andere Programmiersprachen sowie C-Bibliotheksfunktionen in Assemblerprogrammen dürfen nicht verwendet werden, außer dies ist in der Aufgabenstellung ausdrücklich gefordert.

Der Programmmentwurf wird gruppenweise erstellt (max. drei Studierende pro Gruppe). Die Namen der Gruppenmitglieder sind im Quellcode zu vermerken. Außerdem ist zu vermerken, welches Gruppenmitglied welche Aufgabenteile überwiegend bzw. schwerpunktmäßig bearbeitet hat.

Wenn Sie Programmfragmente aus der Literatur, dem Internet oder von anderen Quellen verwenden, ist die Quelle als Kommentar kenntlich zu machen.

Bewertung und Abgabe

Der Aufgaben werden anhand der folgenden Kriterien bewertet:

- Funktionalität, Korrektheit und Effizienz
- Verständlichkeit, Kommentierung, Strukturierung

Der Programmmentwurf ist jeweils einmal pro Gruppe als „gezipptes“ Archiv im tar-Format bis spätestens

Donnerstag, 14. April 2022, 23:59 CEST

per Email an `rdrcode@gmx.eu` sowie in Kopie an das Sekretariat Informationstechnik, Fr. Schmidt zu schicken. Abzugeben sind der vollständige Quellcode, das Makefile sowie das ausführbare Programm.

Beachten Sie bitte, dass Sie für die Vollständigkeit und Lesbarkeit des abgegebenen Quellcodes verantwortlich sind. Der Name der abzugebenden Archivdatei ist `pe_tit20_nachname1_nachname2_nachname3.tar.gz`.

Aufgabenstellung

Schreiben Sie ein Assembler-Programm `timediff`, welches eine Folge aufsteigend sortierter Zeitstempel einliest und für jeden Zeitstempel die Zeitdifferenz zu dessen Vorgänger ausgibt.

- i. Die Eingabe erfolgt zeilenweise von der Standardeingabe als formatierter ASCII-Text.
- ii. Gültige Eingabezeilen enthalten genau einen Zeitstempel.

- iii. Eingabezeilen werden durch einen Zeilenumbruch abgeschlossen.
- iv. Die Ausgabe erfolgt auf der Standardausgabe. Die Anzahl der System-Call Aufrufe (hier System-Call `write`) ist zu minimieren.
- v. Eingabezeilen werden nicht ausgegeben.
- vi. Die Ausgabe erfolgt erst, nachdem der Eingabetext vollständig eingelesen wurde.
- vii. Ein Eingabetext kann maximal 10000 Zeitstempel enthalten.
- viii. Ein gültiger Zeitstempel wird im Format `S+.M+` angegeben, wobei `S` die Anzahl Sekunden seit der UNIX Epoche angibt und `M` die Anzahl Mikrosekunden seit Sekundenbeginn. `S` und `M` sind Dezimalziffern im ASCII-Format.
- ix. Ein Zeitstempel ist in eine Struktur des Typs `struct timeval` zu konvertieren (Definition siehe `man 2 gettimeofday`).
- x. Wenn eine Eingabezeile einen ungültigen Zeitstempel enthält, dann beendet sich das Programm mit einer entsprechenden Fehlermeldung.
- xi. Der Sekundenanteil eines Zeitstempels ist bei der Eingabe vorzeichenlos und mindestens einstellig mit einem maximalen Wertebereich von 64 Bit.
- xii. Der Mikrosekundenanteil eines Zeitstempels ist bei der Eingabe grundsätzlich auf sechs Stellen zu normieren, d.h. der Mikrosekundenanteil des Zeitstempels `1502736311.5` ist `500000`.
- xiii. Die konvertierten Zeitstempel werden in einer Liste gespeichert.
- xiv. Das Modul `Liste` implementiert mindestens die in der Datei `list.asm` vordefinierten Funktionen.
- xv. Die Datei `list_test.c` definiert den Modultest für das Modul `Liste`.
- xvi. Für die Ausgabe werden die Zeitstempel aus der Liste ausgelesen.
- xvii. Die Ausgabe der Zeitstempel und der Zeitdifferenzen muss dem Format des unten aufgeführten Beispiels entsprechen.

Beispiel einer Eingabefolge:

```
1000000000.0
1234567890.000000
1483225200.000000
1491861600.000
1500000000.000000
1502529000.000001
1502529001.000000
1502530860.999999
1502617201.999998
```

```
1502617202.000000
1502736311.000001
```

Format für die Ausgabe:

```
1000000000.000000
=====
1234567890.000000
2714 days, 20:44:50.000000
=====
1483225200.000000
2877 days, 23:28:30.000000
=====
1491861600.000000
100 days, 00:00:00.000000
=====
1500000000.000000
94 days, 04:40:00.000000
=====
1502529000.000001
29 days, 06:30:00.000001
=====
1502529001.000000
00:00:00.999999
=====
1502530860.999999
00:30:59.999999
=====
1502617201.999998
23:59:00.999999
=====
1502617202.000000
00:00:00.000002
=====
1502736311.000001
1 day, 09:05:09.000001
```