

1 Aufgabe 4 Binäre Suche

1.1 Hintergrund

Die binäre Suche ist ein Algorithmus, der auf einem Array recht schnell ein gesuchtes Element findet bzw. eine zuverlässige Aussage über das Fehlen dieses Elementes liefert. Voraussetzung ist, dass die Elemente des Arrays in einer dem Suchbegriff entsprechenden Weise sortiert sind. Der Algorithmus basiert auf dem Schema Teile und Herrsche.

1.2 Verfahren

Zuerst wird das mittlere Element des Arrays überprüft. Es kann kleiner, größer oder gleich dem gesuchten Element sein. Ist es kleiner als das gesuchte Element, muss das gesuchte Element in der hinteren Hälfte stecken, falls es sich dort überhaupt befindet. Ist es hingegen größer, muss nur in der vorderen Hälfte weiter gesucht werden. Die jeweils andere Hälfte muss nicht mehr betrachtet werden. Ist es gleich dem gesuchten Element, ist die Suche (vorzeitig) beendet. Jede weiter zu untersuchende Hälfte wird wieder gleich behandelt: Das mittlere Element liefert wieder die Entscheidung darüber, wo bzw. ob weiter gesucht werden muss. Die Länge des Suchbereiches wird von Schritt zu Schritt halbiert. Spätestens wenn der Suchbereich auf 1 Element geschrumpft ist, ist die Suche beendet. Dieses eine Element ist entweder das gesuchte Element, oder das gesuchte Element kommt nicht vor. Der Algorithmus zur binären Suche wird entweder als Iteration oder Rekursion implementiert

1.3 Aufgabe

Bitte schreiben Sie ein Programm, das einen Array von Strings nach einem bestimmten String durchsucht. Das Array ist Alphabetisch sortiert.

1.3.1

Es soll eine Funktion geschrieben werden, die zwei Strings vergleicht.

Parameter in:

Die Adressen der beiden Strings

Parameter out:

0 Die Strings sind identisch

+1 Der erste String ist größer als der zweite (z.B. Hugo > Huga)

-1 Der erste String ist kleiner als der zweite (z.B. Anke < Anne)

1.3.2

Die Suche soll zunächst als lineare Suche realisiert sein. (ARM Procedure Call Standard!) Dabei soll die Funktion aus 1.3.1 benutzt werden

Parameter in:

Adresse des gesuchten Strings

Adresse des Arrays

Anzahl der Elemente im Array

Größe eines Elements.

Parameter out:

0 Element nicht gefunden

sonst Adresse des Elements im Array

1.3.3

Danach soll die Suche als binäre Suche realisiert werden.

Parameter wie bei 1.3.2

1.3.4

Als letztes soll die binäre Suche als rekursives Unterprogramm implementiert werden.

Parameter wie bei 1.3.2

Protokoll:

Quellcode von allen Versionen

Screenshots von Testergebnissen für all 6 Testnamen (bei Rekursion) Speicherauszug, der die Stack Entwicklung bei der Rekursion dokumentiert.

Vergleichen Sie bitte die Laufzeit der Rekursion mit der der linearen Suche wenn Ihnen noch Zeit bleibt.

1.4 Vorzubereiten

Unterprogrammtechniken mit dem ARM Assembler, (Aufruf, Rücksprung, Parameter, Rückgabewerte), Konzeption der Rekursion.