

Aufgabe 2:

wsort

a) Standard-Eingabe sortieren

Schreiben Sie ein Programm `wsort`, das eine Liste von Wörtern (jedes Wort steht in einer eigenen Zeile!) vom Standard-Eingabekanal (`stdin`) einliest, diese Liste alphabetisch sortiert und die sortierte Liste auf dem Standard-Ausgabekanal (`stdout`) wieder ausgibt (wieder ein Wort pro Zeile).

In den Dateien `~i4sp/pub/aufgabe2/wlist*` finden Sie Beispiel-Eingabedateien. Mit dem Programm `sort(1)` können Sie diese Dateien sortieren und die Ausgabe jeweils mittels `diff(1)` mit der Ausgabe Ihres `wsort`-Programms vergleichen.

Zum Sortieren in Ihrem Programm verwenden Sie die Funktion `qsort(3)` aus der ANSI-C-Bibliothek. Mit der Funktion `strcmp(3)` können Sie den alphabetischen Vergleich zweier Worte durchführen. Mit Hilfe von `malloc(3)` können Sie dynamisch Speicher nachfordern, um Datenstrukturen anzulegen, die Sie zum Sortieren benötigen.

b) Argumente aus der Kommandozeile sortieren

Ergänzen Sie das Programm nun so, daß es die Wörter, die als Übergabeparameter übergeben werden, sortiert und auf `stdout` ausgibt. Werden keine Parameter übergeben, soll sich das Programm wie unter (a) verhalten.

Hinweise:

Das C-Programm ist in der Datei **`wsort.c`** abzulegen. Unterprogramme und Variablendefinitionen sind ausreichend zu kommentieren. Achten Sie bitte außerdem auf saubere Gliederung des Quellcodes! Sie können davon ausgehen, daß die Wörter eine maximale Länge von 100 Zeichen nicht überschreiten. Anderenfalls soll eine Fehlermeldung ausgegeben und das Wort ignoriert werden. Leere Wörter sollen ohne eine Fehlermeldung ignoriert werden.

In einer Datei **`wsort.doc`** in dem Directory `aufgabe2/doc` ist außerdem die Vorgehensweise und der Aufbau der Datenstrukturen zum Aufruf von `qsort()` knapp aber präzise und vollständig zu beschreiben. (Nicht der Quick-Sort-Algorithmus selbst, sondern was man mit den Eingabedaten machen muß, um die `qsort`-Funktion mit Daten zu versorgen).

Abgabe: bis spätestens Mittwoch, 21.11.2001, 20:00 Uhr