Aufgabe 2: wsort (12.0 Punkte)

a) Standard-Eingabe sortieren
Schreiben Sie ein Programm `wsort` in einer Datei `wsort.c`, welches eine Liste von Wörtern (jedes Wort steht in einer eigenen Zeile, **fgets(3)**) vom Standard-Eingabekanal (*stdin*) einliest, diese Liste alphabetisch sortiert (**qsort(3)**, **strcmp(3)**) und die sortierte Liste auf dem Standard-Ausgabekanal (*stdout*) wieder ausgibt (ein Wort pro Zeile). Im Verzeichnis `/proj/i4sp1/pub/aufgabe2` finden Sie Beispiel-Eingabedateien (`wlist*`) sowie eine Vergleichsimplementierung (`wsort`), mit der Sie die Wortlisten sortieren und die Ausgabe jeweils mittels **diff(1)** mit der Ausgabe Ihres eigenen wsort-Programms vergleichen können. Mit Hilfe von **malloc(3)** und **realloc(3)** können Sie dynamisch Speicher an- und nachfordern, um die benötigten Datenstrukturen anzulegen bzw. zu erweitern.

b) Kommandozeilenargumente sortieren
Ergänzen Sie das Programm nun so, dass es die Argumente, die an `wsort` übergeben werden, sortiert und auf *stdout* ausgibt. Werden keine Argumente übergeben, soll sich das Programm wie unter a) verhalten, ansonsten werden nur die Argumente sortiert, keine Daten von der Standardeingabe.

c) Makefile schreiben
Erstellen Sie ein zur Aufgabe passendes Makefile, welches keine ins **make(1)**-Programm eingebauten Makros und Regeln voraussetzt. Das Makefile sollte mit dem Aufruf `make -Rr` funktionieren und die Standardtargets `all` und `clean` mit der üblichen Funktionalität bereitstellen. Außerdem soll ein Target `wsort-debug` vorhanden sein, welches ein wsort-Binary (Dateiname `wsort-debug`) mit Debugsymbolen erzeugt.

Hinweise zur Aufgabe:

- Erforderliche Dateien: `wsort.c, Makefile`
- Ein Wort enthält alle Zeichen einer Zeile. Zeilen sind durch Newline-Zeichen (`\n`) voneinander getrennt, die selbst nicht Teil des Wortes sind. Jede Zeile endet mit einem Newline-Zeichen, mit Ausnahme der letzten Zeile. Enthält die letzte Zeile Zeichen, so ist diese ebenfalls als Wort zu behandeln.
- Wörter, die eine maximale Länge von 100 Zeichen überschreiten, werden mit einer entsprechenden Fehlermeldung ignoriert.
- Leere Zeilen sind ohne Fehlermeldung zu ignorieren.
- Beim Sortieren von großen Dateien hängt es vom Aufbau des Programms ab, wie schnell `wsort` die Daten sortiert. Sie können mit dem Programm **time(1)** die benötigte Ausführungszeit messen und mit anderen Lösungen vergleichen.
- Sämtliche Fehlermeldungen sollen auf dem Standardfehlerkanal (*stderr*) erfolgen. Auf die Standardausgabe (*stdout*) soll ausschließlich die sortierte Wortliste ausgegeben werden, damit die Ausgabe mit anderen Lösungen vergleichbar ist.

Hinweise zur Abgabe:

Bearbeitung: Zweiergruppen
Bearbeitungszeit: 11 Werktage
Abgabezeit: 17:30 Uhr