

## Aufgabe 2:

### wsort (12 Punkte)

Die Aufgabe ist einzeln zu bearbeiten! **Keine Zweiergruppen!**

#### a) Makefile

Erstellen Sie ein einfaches Makefile. Durch den Aufruf von “make” soll Ihr Program übersetzt werden, mit “make install” das übersetzte Programm aus dem Verzeichnis src in das Verzeichnis bin.i386 kopiert werden und mit “make clean” sollen alle durch den Aufruf von “make” erzeugten Dateien im Verzeichnis src gelöscht werden.

#### b) Standard-Eingabe sortieren

Schreiben Sie ein Programm **wsort**, welches 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 /proj/i4sos/pub/aufgabe2/wlist\* finden Sie Beispiel-Eingabedateien. Mit dem Programm /proj/i4sos/pub/aufgabe2/fsort 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 die benötigten Datenstrukturen anzulegen.

#### c) Argumente aus der Kommandozeile 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 (b) verhalten.

#### d) Dokumentation

In einer Datei **wsort.txt** in dem Verzeichnis 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).

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 Zeilen sind ohne Fehlermeldung zu ignoriert.

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.

**Abgabe: bis spätestens Donnerstag, 10.05.2007, 12:00 Uhr**