

# OPERATING SYSTEMS BEISPIEL 1

## Aufgabenstellung A – mysort

Schreiben Sie eine abgewandelte Version des UNIX-Kommandos `sort` als C-Programm. Es soll nur die Option `-r` (absteigend sortieren) implementiert werden. Alle anderen Optionen können ignoriert werden. Wird keine Datei angegeben, so sollen die Daten über *stdin* eingelesen werden.

### SYNOPSIS

```
mysort [-r] [file1] ...
```

## Anleitung

Lesen Sie alle Dateien zeilenweise in einen dafür geeigneten Puffer ein. Sortieren Sie danach die Daten mit Hilfe von *qsort(3)*. Im Anschluss geben Sie die nun sortierten Daten auf *stdout* aus. Es kann davon ausgegangen werden, dass keine Zeile länger als 1022 Zeichen ist (ohne Newline).

## Testen

Testen Sie Ihr Programm mit verschiedenen Eingaben. Erstellen Sie zum Beispiel eine Testdatei `t1` mit folgenden Zeilen:

```
Priority9 cat
Priority2 ls
Priority7 cat mysort.h
```

Rufen Sie Ihr Programm dann wie folgt auf:

```
$ ./mysort < t1
Priority2 ls
Priority7 cat mysort.h
Priority9 cat
```

```
$ cat t1 | ./mysort
Priority2 ls
Priority7 cat mysort.h
Priority9 cat
```

```
$ ./mysort -r t1 t1
Priority9 cat
Priority9 cat
Priority7 cat mysort.h
Priority7 cat mysort.h
Priority2 ls
Priority2 ls
```

## Hinweis

( `echo a ; echo B` ) | `sort` sortiert B vor a, weil die Großbuchstaben im Zeichensatz vor den Kleinbuchstaben kommen. ( `echo a ; echo B` ) | `sort -f` würde diese Eigenschaft ignorieren. Die Opti-

on `-f` muss aber **nicht** implementiert werden.

## Richtlinien

Bitte beachten Sie auch die *Richtlinien für die Erstellung von C-Programmen* auf der Übungswebsite.