

Blatt 08

Abgabe bis: 22. Dezember 2011

11. reverse.c (10P)

Schreibt das Programm `reverse`, das die Reihenfolge der Buchstaben auf der Zeile umdreht.

```
reverse [-c WORD [-x]] [FILES...]
```

- Optionen beginnen mit ‘-’ und stehen vor den Dateiangumenten.
- Werden Dateiangumente angegeben, so sollen sie der Reihe nach abgearbeitet werden. Unter den Dateiangumenten darf auch maximal eines sein, das “-” heisst; an dieser Stelle soll dann die Standardeingabe verarbeitet werden. Wird kein Dateiangument angegeben, dann soll von der Standardeingabe gelesen werden. (vgl. *cat(1)*)
- Alle Ausgabe soll auf die Standardausgabe geschrieben werden. Bei mehreren Dateien sollen die Ausgaben hintereinander gehaengt werden. (vgl. *cat(1)*)
- Die Reihenfolge der zwei Optionen “-c WORD” und “-x” ist beliebig. Aus Kompatibilitaet zu bestehenden Unix-Tools soll sowohl die Notation “-c WORD” als auch “-cWORD” akzeptiert werden.
- Ist “-c WORD” vorhanden, dann sollen nur diejenigen Zeilen umgedreht werden, die den String “WORD” enthalten (*strstr(3)*). Alle anderen Zeilen sollen unveraendert ausgegeben werden.
- Das Flag “-x” invertiert die Filterung von “-c WORD”: Es sollen dann gerade die umgedreht werden, die “WORD” *nicht* enthalten. (“-x” macht scheinbar nur Sinn, wenn “-c WORD” angegeben wurde. Geht damit sinnvoll um.)
- **Wichtig:** Den Code zum Parsen der Optionen muesst ihr selbst schreiben. Verwendet *nicht* die Bibliotheksfunktion *getopt(3)* oder aehnliches.
- **Wichtig:** Zeilen in der Eingabe koennen *beliebig* lang sein. Ihr koennt dafuer keinen Puffer fester Groesse verwenden. Vielmehr muesst ihr ihn dynamisch ver-groessern, falls ueberlange Zeilen auftreten. (Stichwort: *realloc(3)*)
- Gliedert euer Programm in sinnvolle Einheiten, die ihr jeweils in eine Funktion packt. Das ist Pflicht!
- Ihr koennt davon ausgehen, dass alle Eingaben ASCII-Zeichen sind. Ihr muesst euch also nicht um Multibyte-Zeichenkodierungen kuemmern.

Beispiele:

```
$ cat >a
hello
world
^D
$ ./reverse a
olleh
dlrow
$ ./reverse -c el <a
olleh
world
$ cat a | ./reverse -cel -x
hello
dlrow
$ echo foo | ./reverse a - a
olleh
dlrow
oof
olleh
dlrow
$
```

Zum Einreichen eurer Loesung:

```
submit ssl 11 team [notes] reverse.c
```

Fragen zur Selbstkontrolle

- Wie greift man auf Kommandozeilenargumente zu?
- Was ist der Unterschied zwischen `char *argv[]` und `char **argv`?
- Weshalb ist `argv` NULL-terminiert, wo die Laenge doch schon in `argc` steckt?
- Beschreibe kurz die dynamischen Datenstrukturen: Verkettete Liste, Doppelt Verkettete Liste, und Baum. Was bietet C dafuer an?
- Wie implementiert man einen Stack als Verkettete Liste?
- Wie findet man Fehler in Programmen die dynamischen Speicher verwenden? Und wichtiger: Wie geht man vor um sie zu reparieren?
- Was ist ein Backtrace und wie bekommt man sowas?