

# DOKUMENTATION

ZUR AUFGABE 4 AUS DER VORLESUNGSREIHE  
“BETRIEBSSYSTEME”

Anton & Mesut

## Einleitung

Bei dieser Aufgabe geht es um die Entwicklung eines Linux Device Drivers. Die Darstellung orientiert sich an dem SCULL (Simple Character Utility for Loading Localities) Treiber. SCULL arbeitet als Charakter-Treiber mit einem Speicherbereich, der Treiber selbst simuliert ein Gerät mit einem vordefinierten Adressbereich für Funktionseinstellungen und Kommunikationsschnittstellen.

Die Aufgabe ist es einen Gerätetreiber zu entwickeln, der eine Verschlüsselung der Buchstabenreihenfolge bildet.

## Package – Komponente

Folgende Komponenten enthält die Aufgabe:

- Setup (install)
- Übersetzungsalg. (translate)
- Cäsar-Verschlüsselung( caesar)
- Mainfile (main)
- Makefile (Flags können gesetzt werden)

### install.sh

- Shell Script Datei für den Aufruf der der Applikation.

### translate.c

- Beinhaltet zwei Geräte /dev/trans0 und /dev/trans1 für die Verschlüsselung und Entschlüsselung des gegebenen Charakters.
- Beide Geräte werden von ein und dem selbem Modul gesteuert.

### caesar.c

- Verschlüsselung erfolgt nach dem Cäsar-Verschiebungsverfahren.
- Naives und unsicheres Verschlüsselungsverfahren.
- Bei der Verschlüsselung wird jeder Buchstabe von dem Klartext auf den Verschlüsselungstext abgebildet. Die Abbildung findet statt indem man die Zeichen nach einer bestimmten Anzahl zyklisch nach rechts verschiebt. Für die Entschlüsselung wird um die gleiche Anzahl wieder nach links verschoben.
- Mathematische Definition mit Hilfe der Modulo-Addition:

*P = Klartext; K=Verschiebung um K Zeichen; Anzahl an Zeichen des Alphabets = 26;*

Verschlüsselung:  $\mathbf{K(P) = (P+K) \bmod 26}$

*C = Geheimtextbuchstabe;*

Entschlüsselung:  $\mathbf{K(C) = (C-K) \bmod 26}$

- Eine Abbildung für eine mögliche Variante einer Verschlüsselung:

Klar:	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Geheim:	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

### main.c

- Testet unseren Verschlüsselungsalgorithmus, mit einer Hilfe einer vordefinierten Message.

### translate.h

- Enthält Header-Dateien für das Kernel (Config).
- Struktur inkl. Semaphore.

## Beschreibung der Aufgabe

Die Applikation ruft beim Start die Funktion **trans\_open** auf, um die Semaphore zu blockieren. Entweder startet die Prozedur danach, oder sie gibt zurück, dass dieser Semaphore blockiert ist. Das **trans\_close** (*release*) ist dabei das Gegenteil, welches den Semaphoren nach Beendung eines Zugriffes freigibt. Bei dem Device trans0 wird die Eingabe verschlüsselt und der Buffer für die Eingabe ist auf 40 Chars begrenzt. Sollte der Buffer Volllaufen, wird der Zugriff unterbrochen. Dementsprechend wird über dem Device trans1 der Buffer dann wieder decodiert und in den user\_space abgelegt.

## Quellen Angabe

- Jürgen Quade – Eva Katharina Kunst „Linux Treiber entwickeln“
- <https://de.wikipedia.org/wiki/Caesar-Verschl%C3%BCsslung>