

2. Übung

Ausgabe Abgabe
24.10.14 07.11.14

Bitte bei der Abgabe Name der Mitglieder einer Gruppe, Nummer der Übung/Teilaufgabe und Datum auf den Lösungsblättern nicht vergessen! Darauf achten, dass die Lösungen beim richtigen Tutor abgegeben werden. Achten Sie bei Programmieraufgaben außerdem darauf, dass diese im Linuxpool kompilierbar sind. Nutzen Sie dazu die Flags `-std=c99`, `-Wall` und `-pedantic`. Es sollten keine Warnungen auftauchen.

Zu spät abgegebene Lösungen werden nicht berücksichtigt!

Aufgabe 1: Behauptungen über Betriebssysteme (2 Punkte)

Beurteilen Sie den Wahrheitsgehalt der folgenden Aussagen und begründen Sie Ihre Entscheidung. Wenn Sie Spezialfälle kennen, die dem Standardfall widersprechen, so schreiben Sie diese mit auf.

- Ein Microkernel beschränkt sich auf grundlegende Speicherverwaltung und Kommunikation zwischen Prozessen.
- Im Kernelmodus muss das Betriebssystem darauf achten, dass die Speicherbereiche verschiedener Prozesse voneinander isoliert sind.
- Interrupts können nicht unterbrochen werden.
- Bei Microkernen kommt es häufiger zu einem Kontextwechsel als bei monolithischen Kernen.
- Die erste Implementierung von Syscalls (siehe Folie 2.21) ist für ein Microkernel geeignet.
- Die zweite Implementierung von Syscalls (siehe Folie 2.22) ist für einen monolithischen Kernel geeignet.
- In der dritten Implementierung von Syscalls (siehe Folie 2.23) werden Syscalls durch einen Nachrichtenaustausch realisiert.
- In der vierten Implementierung von Syscalls (siehe Folie 2.24) werden Syscalls durch einen Nachrichtenaustausch realisiert.

Aufgabe 2: Einführung in C: Felder (3 Punkte)

Diese Aufgabe behandelt Felder. Felder haben eine feste Länge und können Objekte eines bestimmten Typs aufnehmen. So ist

```
int a[10];
```

ein Feld (bzw. eng. *Array*) von 10 Elementen, die jeweils vom Typ `int` sind. C kennt keinen eigenen Datentyp für Strings. Stattdessen wird ein String durch ein Array mit Elementen vom Typ `char` repräsentiert, das mit dem Zeichen „\0“ endet (Strings sind *null-terminiert*). Ein String vom Typ

```
char s[10];
```

kann also neun Zeichen und das terminierende „\0“ aufnehmen. Strings können in der Variablen Deklaration initialisiert werden, z.B.:

```
char s[10] = "hallowelt";
```

Man kann direkt auf die Elemente eines Arrays zugreifen, indem man ihren Index angibt. Zum Beispiel:

```
a[2] = 13;
```

Dabei ist zu beachten, dass das erste Element eines Feldes den Index 0 hat!

Ein klassischer Verschlüsselungsalgorithmus ist die *Vigenere-Chiffre*. Er funktioniert wie folgt: Man nimmt zunächst den zu verschlüsselnden Klartext, wandelt alle Klein- in Großbuchstaben um und entfernt alle Leer- und Sonderzeichen. Den Buchstaben A bis Z werden die Indizes 0 bis 25 zugeordnet, die Ziffern 0 bis 9 erhalten die Indizes 26 bis 35. Dann schreibt man unter jeden Buchstaben des Textes einen Buchstaben vom Codewort und addiert die Indizes, z.B. B (1) + E (4) ergibt F (5) und 2 (28) + U (20) ergibt M (48 modulo 36 = 12). Ist der Text länger als das Codewort, wird das Codewort mehrfach wiederholt.

```
Beispiel: Codewort SYSPR006
Klartext : Dieser Text soll verschluesselt werden.
```

```
Codierung: DIESERTEXTSOLLVERSCHLUESSELTWERDEN
Codewort  : SYSPR006SYSPR006SYSPR006SYSPR006SY
-----
Chiffre   : V6W7V5JAFHA32ZLA9GUW2840A238DSH9WB
```

Der Empfänger der Nachricht kann den Text entschlüsseln, indem er unter den chiffrierten Text wieder das Codewort schreibt, diesmal die Indizes jedoch subtrahiert:

```
Chiffre   : V6W7V5JAFHA32ZLA9GUW2840A238DSH9WB
Codewort  : SYSPR006SYSPR006SYSPR006SYSPR006SY
-----
Codierung: DIESERTEXTSOLLVERSCHLUESSELTWERDEN

Klartext  : Dieser Text soll verschluesselt werden.
```

- Schreiben Sie ein Programm, das vom Benutzer zunächst ein Codewort (max. 10 Zeichen) und dann einen beliebigen Text (max. 100 Zeichen) erfragt. Fragen Sie ab, ob der Text ver- oder entschlüsselt werden soll. Führen Sie die Vorverarbeitung des Textes sowie die Ver-/Entschlüsselung durch und geben Sie das Ergebnis auf dem Bildschirm aus.
- Worin könnten Schwächen des *Vigenere-Chiffres* bestehen? Begründen Sie Ihre Vermutung.